
BACHELORARBEIT

Frau
Jasmin Rosenfelder

**Analyse und Auswahl geeig-
neter Add-ons zur computer-
gestützten forensischen
Tatortrekonstruktion mittels
Open-Source-Software**

2016

BACHELORARBEIT

Analyse und Auswahl geeigneter Add-ons zur computergestützten forensischen Tatortrekonstruktion mittels Open-Source-Software

Autor/in:

Frau Jasmin Rosenfelder

Studiengang:

Medieninformatik und interaktives Entertainment

Seminargruppe:

MI13w2-B

Erstprüfer:

Prof. Dr. rer. nat. Dirk Labudde

Zweitprüfer:

M. Sc. Sven Becker

BACHELOR THESIS

Analysis and Selection of suitable Add-ons for the digital forensic Crime Scene Reconstruction through Open Source Software

author:

Ms. Jasmin Rosenfelder

course of studies:

Mediainformatics and interactive Entertainment

seminar group:

MI13w2-B

first examiner:

Prof. Dr. rer. nat. Dirk Labudde

second examiner:

M. Sc. Sven Becker

Bibliografische Angaben

Nachname, Vorname: Rosenfelder, Jasmin

Analyse und Auswahl geeigneter Add-ons zur computergestützten forensischen Tatort
Rekonstruktion mittels Open-Source-Software

72 Seiten, Hochschule Mittweida, University of Applied Sciences,
Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2016

Referat

Ziel der vorliegenden Bachelorarbeit war es, ein Open-Source-Softwarepaket zusammen zu stellen, das zur computergestützten Tatortrekonstruktion verwendet werden kann. Dazu wurden verschiedene Add-ons für die Open-Source-Software Blender in ihrer Funktionsweise vorgestellt und anschließend miteinander verglichen. Hierbei wurde insbesondere auf Nutzerfreundlichkeit und Eignung zur Tatortrekonstruktion eingegangen. Anschließend wurde das daraus resultierende Softwarepaket zur Erstellung einer Tatortrekonstruktion anhand eines Fallbeispiels verwendet.

Abstract

The goal of this Bachelor's thesis is to create an open source software package, which is suitable for digital crime scene reconstruction. To accomplish this, different add-ons available for the open source software Blender are introduced in detail and compared with each other. The focus of this process is user-usability and the suitability of the add-ons in terms of crime scene reconstruction. Subsequently, the software package, resulting from the analysis, is used to construct an exemplary crime scene.

Inhaltsverzeichnis

Inhaltsverzeichnis.....	V
1 Einleitung.....	1
1.1 Motivation.....	1
1.2 Problemstellung	1
1.3 Zielsetzung.....	2
2 Theoretische Grundlagen	3
2.1 Tatortrekonstruktion	3
2.1.1 Definition.....	3
2.1.2 Klassische Aufnahmetechniken.....	3
2.1.3 Computergestützte Aufnahmetechniken	4
2.2 Open-Source Software Blender.....	4
2.3 Erweiterung der Software	6
2.3.1 Add-on	6
2.3.2 Standalone	6
3 Hintergründe zum ausgewählten Fallbeispiel.....	7
4 Analyse verschiedener Add-ons.....	8
4.1 Tatorterstellung	8
4.2 Personenerstellung	16
4.3 Visualisierung von Bewegungsmustern	23
4.4 Simulieren der Tathergänge	32
5 Auswahl geeigneter Add-ons.....	43
6 Anwendung geeigneter Add-ons auf ein Fallbeispiel	45
7 Diskussion.....	51
8 Zusammenfassung und Ausblick.....	53
Literaturverzeichnis.....	VI
Anlagen	IX
Eigenständigkeitserklärung	XIX
.....	

1 Einleitung

1.1 Motivation

Der Tatort eines Verbrechens kann als wichtigster Zeuge angesehen werden, da die darin gefundenen Daten maßgebend zur Aufklärung des Verbrechens beisteuern können. Jedoch ist die Untersuchung und Aufbereitung der gesammelten Daten eines Tatortes ein komplexer Prozess. Hierzu müssen beispielsweise verschiedene Fotografien, Protokolle und Messdaten aufbereitet werden, um anschließend den Ermittlern zu helfen, das Verbrechen aufzuklären [Kw07].

Durch die fortschreitende Entwicklung der Technik wurde dieser Prozess zunehmend erleichtert, so war es ab den 70iger Jahren beispielsweise möglich, computergestützte 3D-Modelle zu erstellen und zu verarbeiten. [CI12] Die Visualisierung des dreidimensionalen Raumes ebnete den Weg zu neuen Möglichkeiten in unterschiedlichsten Feldern, unter anderem auch im Bereich der Kriminalistik.

Mittels 3D-Programmen wie 3Ds-Max [URL-1] ist eine computergestützte Tatortrekonstruktion realisierbar. Hierbei wird entweder händisch der Tatort maßstabsgetreu nachmodelliert, oder es werden die Daten eines 3D-Scanners mit Hilfe entsprechender Software direkt als 3D-Modell ausgegeben. [FI11] Durch die Verwendung solcher Technologien kann eine repräsentative Veranschaulichung des Tatortes realisiert werden – eine solche Veranschaulichung kann nicht nur den Ermittlern selbst Aufschluss über den Tathergang erschaffen, sondern auch maßgebend vor Gericht benutzt werden. Außerdem ist eine 3D-Rekonstruktion hilfreich, wenn der Tatort entweder nicht zugänglich ist, weil sonst Spuren zerstört werden könnten, beziehungsweise weil der Tatort aus Sicherheitsgründen nicht betreten werden darf, oder aber nicht mehr existiert, da eine Wiederaufnahme eines Falles stattfindet, bei der der Tatort z.B. abgerissen wurde. Hier kann durch die virtuelle Tatortrekonstruktion anhand von Skizzen, Messdaten und Fotografien nicht nur eine visuelle Veranschaulichung stattfinden, sondern dies kann auch Aufschluss über den Tathergang bringen. Dies könnte sich ohne einen „virtuell begehbaren“ Tatort als schwierig erweisen.

1.2 Problemstellung

Da die 3D-Rekonstruktion eines Tatortes zunehmend wichtiger und alltäglicher am kriminalistischen Arbeitsplatz geworden ist, gibt es verschiedenste Programme, die zu diesem Zweck verwendet werden können. So wird kommerzielle Software wie 3Ds-Max benutzt, um den Tatort zu modellieren. 3Ds-Max ist jedoch ein 3D-Modelling Pro-

gramm, das hauptsächlich für die Erstellung von Modellen für die Unterhaltungsbranche erstellt wurde. Die Schwerpunkte der Software liegen bei der Modellierung, Animation und Texturierung von 3D-Objekten. Ein Nachteil bei der Verwendung von kommerzieller und nicht für den forensischen Gebrauch entwickelter Software ist der hohe Kosten- und Zeitaufwand, der nötig ist, um ein repräsentatives 3D-Modell zu erstellen. [CI12] Obwohl die Software zunehmend benutzerfreundlich wird, braucht es dennoch einige Schulungen, um sich den entsprechenden Erfahrungsschatz für eine komplexe Aufgabe, wie die Modellierung eines Tatortes, anzueignen. Des Weiteren kann die Software nicht in jedem Bereich der Tatortrekonstruktion verwendet werden. Eine Modellierung des Tatortes ist möglich, doch sind weitergehende Prozesse wie beispielsweise physikalische Simulationen nur durch den Erwerb kostspieliger Plugins möglich.

Neben Software wie 3Ds-Max kann aber auch 3D-Software verwendet werden, die die Daten eines 3D-Scanners auslesen kann. Ein bekanntes Beispiel ist hier das GOM-TRITOP/ATOS-System [URL-2]. Diese sind jedoch an den 3D-Scanner gebunden, und auch dies ist mit teils hohen Kosten verbunden.

Eine weitere flexiblere Lösung ist auf Open-Source Software wie Blender [URL-3] auszuweichen, diese ist kostenlos und verfügt über eine große Auswahl an Add-ons, welche ebenfalls kostenfrei zur Verfügung gestellt werden.

1.3 Zielsetzung

Durch die Auswahl und Analyse geeigneter Add-ons für die Open-Source Software Blender soll eine umfangreiche Rekonstruktion eines Tatortes ermöglicht werden. Hierzu wird Blender als Grundbaustein verwendet und durch verschiedene Add-ons erweitert. Diese sollen auf Nutzen geprüft werden und anschließend anhand eines Fallbeispiels zeigen, wie die Tatortrekonstruktion durch Open-Source-Tools vereinfacht und verbessert werden kann. Dabei sollen möglichst alle Bereiche der Tatortrekonstruktion abgedeckt werden. Hierbei wird systematisch vorgegangen und zuerst der Tatort, d. h. das Gebäude oder der abgrenzbare Raum, der als Tatort deklariert wurde, mittels eines Add-ons weitestgehend automatisiert erstellt. Anschließend sollen Täter und Opfer ihren äußerlichen Merkmalen nach kreiert werden. Eventuelle Bewegungsabläufe und der Tathergang sollen durch die Benutzung von Add-ons schließlich die Tatortrekonstruktion abrunden.

Ziel ist es, mit Hilfe verschiedener Add-ons eine komplexe Open-Source-Lösung zu erstellen, die einen Gesamteindruck des Tatortes, des Tatherganges und der darin involvierten Personen zulässt.

2 Theoretische Grundlagen

2.1 Tatortrekonstruktion

2.1.1 Definition

Tatortrekonstruktion kann grundlegend von zwei unterschiedlichen Standpunkten betrachtet und definiert werden. Bevor es möglich war, einen Tatort mit entsprechender Software virtuell zu erstellen, beschrieb der Begriff Tatortrekonstruktion die Rekonstruktion des Tatherganges. Hierbei werden alle physisch gesicherten Spuren, die am Tatort gefunden wurden, untersucht und interpretiert. Dabei werden verschiedenen Theorien zum Tathergang herangezogen und mit den Spuren abgeglichen, um die Tat möglichst exakt zu rekonstruieren und somit das Verbrechen bzw. den Unfall aufzuklären. [Fl11]

Durch die Verwendung von 3D-Software ist es möglich die Tatortrekonstruktion auch computergestützt auszuführen. Hierbei wird der Tatort und alle relevanten Spuren entweder exemplarisch maßstabsgetreu erstellt oder auch mit Hilfe von 3D-Scans, Fotografie und Videografie nahezu originalgetreu virtuell wiedergegeben. [Fl11] Dadurch ist es möglich den Tatort zu untersuchen ohne dabei vor Ort zu sein, oder die Spuren zu zerstören.

2.1.2 Klassische Aufnahmetechniken

Bei der Aufnahme eines Tatortes werden verschiedene Techniken verwendet, um diesen zu dokumentieren. So werden alle Spuren detailreich protokolliert, wobei auf Größe, Lage und Aussehen der Spuren eingegangen wird. Der Tatort selbst wird vermessen und ebenfalls protokolliert. Um den Tatort wiederzugeben, werden Skizzen mittels Stift und Papier angefertigt, die beispielsweise den Aufbau einer Wohnung und wichtige Spuren darin darstellen. Ein wichtiger Teil der Tatortaufnahme ist, neben der Protokollierung des Tatortes, auch die Fotografie. Jede Spur wird so fotografiert, dass Lage, Größe und Form der Spur ersichtlich wird. Dies ist möglich, indem Fotografien aus verschiedenen Winkeln erstellt werden, und durch die Verwendung von Maßstäben, die neben den Spuren platziert werden. Neben den Planskizzen des Tatortes werden auch hierzu Fotografien, teils auch Videografien, verwendet, damit ein komplexer Gesamteindruck des Tatortes entstehen kann. [St08]

2.1.3 Computergestützte Aufnahmetechniken

Seit den 30er Jahren werden in der Forensik photogrammetrische Messsysteme und seit den 90er Jahren die Nahbereichsphotogrammetrie verwendet. Diese kann zur Vermessung des Tatortes sowie zur Erstellung von 3D-Plänen benutzt werden. [Bu10] Hierbei werden mehrere Fotos des Tatortes aus verschiedenen Winkeln geschossen und mittels entsprechender Software anschließend in ein 3D-Objekt umgewandelt, dies kann mit speziellen Kameras realisiert werden, aber auch durch die Verwendung einer herkömmlichen Digitalkamera. Dadurch ist nicht nur eine genaue Visualisierung des Tatortes möglich, sondern auch eine genaue Vermessung. Eine weitere gängige Messmethode ist der 3D-Scanner. Dieser tastet die Oberfläche des Tatortes ab und erstellt aus diesen Daten ebenfalls ein 3D-Objekt. Andere Messmethoden sind die Tachymetrie und Messung mittels GPS. [Bu10] Diese Methoden haben händische Vermessung mittels eines Meterstabes mittlerweile fast vollständig abgelöst, da die neueren Methoden nicht nur schneller, sondern auch genauer sind. Des Weiteren ermöglicht Laser-Scanning die Erfassung kleinster Objekte und Verformungen der Oberfläche. Beispielsweise kann ein 3D-Scanner Fußabdrücke erfassen, die dann mittels Software weiterverarbeitet werden können. [Th05] So ist es möglich, die Spuren zu sichern, ohne dabei diese zu berühren und somit eventuell zu schädigen.

2.2 Open-Source Software Blender

Blender ist eine von der Blender Foundation herausgegebene 3D-Suite, die sowohl von der Foundation selbst, als auch den Anwendern, fortlaufend entwickelt wird. Die Software ist nicht nur in der Lage 3D-Objekte zu erstellen, sondern diese auch zu texturieren, riggen und animieren. Zudem ist es möglich eine fertige 3D-Szene zu rendern und anschließend per Compositing optisch zu modifizieren. Durch die integrierte Game-Engine können 3D-Spiele vollständig in Blender erstellt werden. Auch physikalische Simulationen und Motion Tracking können mit Blender realisiert werden. [St12]

Das Programm ist für Linux, Mac und Windows verfügbar und ist unter der „GNU General Public License“ (GPL) lizenziert. Da Blender eine Open-Source-Software ist, steht es jedem frei, die Software für jegliche Zwecke zu benutzen und zu verändern. Dies ist z.B. durch die Verwendung von Pythonskripten oder Add-ons für die Software möglich. [St12]

Durch die vielfältigen Anwendungsmöglichkeiten der Software steht sie den kommerziellen 3D-Programmen wie 3ds-Max oder Maya in nichts nach und wird immer häufiger in der Animations- und Videospielbranche sowie in der Forschung verwendet.

Die Software lässt sich unter [URL-3] herunterladen und anschließend auf den oben genannten Betriebssystemen installieren. Anschließend soll die Oberfläche Blenders und einige Grundfunktionen zum besseren Verständnis erklärt werden.

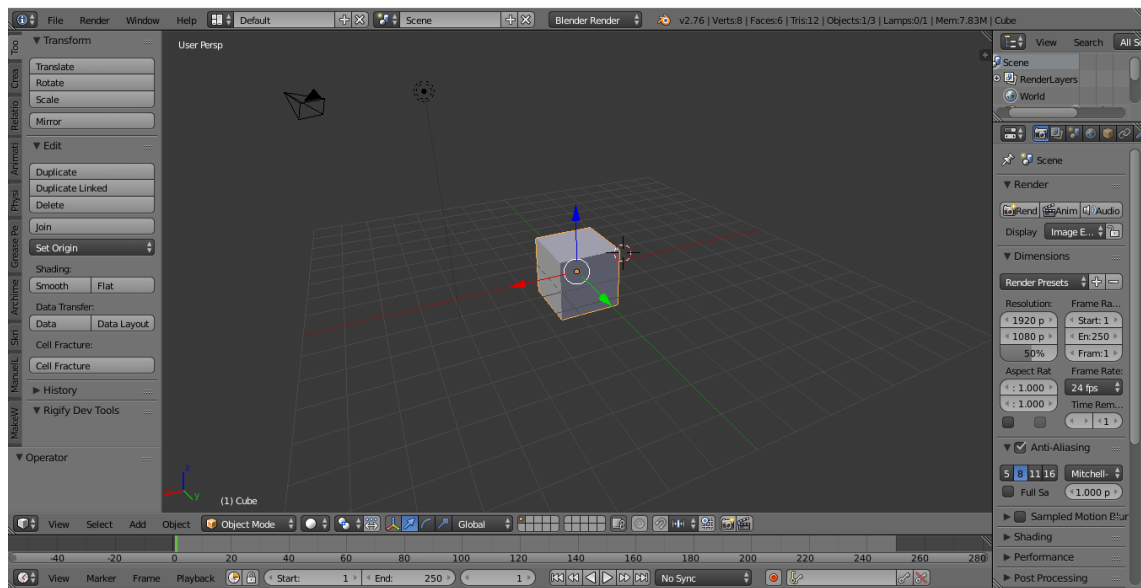


Abbildung 1: Bildschirmfoto der Oberfläche von Blender 2.76

Auf Abb. 1 ist ein Würfel in der Mitte des Bildschirms zu sehen. Dieser Raum wird als 3D-Viewport [Ca15] bezeichnet. Um die Perspektive der 3D-Szene zu ändern, wird das Scrollrad der Maus gedrückt und die Maus bewegt. Um in die Szene hinein- bzw. heraus zu zoomen wird das Scrollrad gedreht. Soll das Objekt bewegt werden, wird es mit der rechten Maustaste ausgewählt und kann nun entlang der x-, y- und z-Achse verschoben werden. Hierzu wird auf einen der Pfeile geklickt und das Objekt in die entsprechende Richtung gezogen.

Auf der linken Seite zum 3D-Viewport befindet sich das Tool-Panel, in dem sich Werkzeuge, mit denen sich die ausgewählten Objekte in der 3D-Ansicht verändern lassen können. [St13] Beispielsweise lässt sich mit Buttons wie „Scale“, „Transform“ oder „Rotate“ zwischen Skalieren, Transformieren und Rotieren des Objektes wechseln.

Zur rechten Seite des 3D-Viewortes befindet sich der Outliner, indem alle Objekte der Szene schematisch aufgereiht sind, unterhalb dessen befindet sich das Property-Panel. [St13] Dieses beinhaltet die Möglichkeit, verschiedene Eigenschaften der Szene und der Objekte zu konfigurieren.

2.3 Erweiterung der Software

2.3.1 Add-on

Ein Add-on ist ein Softwarepaket das zu einer anderen Software ergänzend wirkt. Das Softwarepaket kann nur durch die Benutzung des dafür vorgesehenen Programmes verwendet werden, und muss meist installiert werden.

Durch die Verwendung von Add-ons kann Blender fast unbegrenzt erweitert werden. Es ist entweder möglich, selbst eine Erweiterung für die Software zu schreiben, oder auf schon bestehende Add-ons der Community zurückzugreifen. Alle Add-ons sind kostenfrei erhältlich. Ein Beispiel für ein Add-on wäre das von Manuel Bastioni erstellte „Manuel Bastioni Lab“ [URL-4]. Dieses Add-on ist für die Erstellung von Menschen als 3D-Objekt entwickelt worden. Um es in Blender zu integrieren, muss es heruntergeladen und anschließend in Blender installiert werden. Je nach Add-on verändert sich daraufhin das äußere Erscheinungsbild des Programms.

Da Blender der „General Public Licence“ (GPL) unterliegt, kann Blender in jeder Weise modifiziert werden. So ist es dem Anwender möglich, eine Vielzahl von Add-ons zu installieren und dadurch ein völlig anderes Nutzererlebnis zu erfahren.

2.3.2 Standalone

Ein Standalone ist ein Programm, dass ohne die Verwendung zusätzlicher Programme funktioniert.

Da Blender alle gängigen 3D-Objektformate unterstützt, kann Blender auch durch andere Programme erweitert werden. Dies können andere 3D-Programme wie 3Ds-Max sein, aber auch weniger bekannte Standalones wie das Programm „MakeHuman“ [URL-5] können zusammen mit Blender komplementär verwendet werden. „MakeHuman“ ist wie Manuel- Bastioni-Lab ein Tool zur Entwicklung von Personen im virtuellen Raum. Im Gegensatz zu Manuel-Bastioni-Lab kann MakeHuman unabhängig von Blender verwendet werden. Nicht alle Standalones sind wie Blender kostenfrei, jedoch gibt es auch hier, wie z.B. MakeHuman, ebenfalls Open-Source-Tools, die jeder frei verwenden kann.

3 Hintergründe zum ausgewählten Fallbeispiel

Bei dem ausgewählten Fallbeispiel handelt es sich um einen Doppelmord aus dem Jahre 1964. Dieser wurde aus dem Buch „Tatorte 2“ von Karsten Schinzling und Thomas Schade entnommen. Bei diesem Fall wurde das Ehepaar Elsa und Paul Tomschke von dem ehemaligen Mitarbeiter Klaus Schuricht in ihrer Gaststätte „Hirsch“ in Radeberg ermordet.

Der Täter versteckte sich nach Ladenschluss in der Gaststätte und überwältigte zuerst die Frau, die aus der Küche im Erdgeschoss kam, indem er ihr mit einem Hammer in den Kopf schlug, und als sie auf den Boden stürzte noch mehrmals nach ihr trat. Danach kam es zu einem Handgemenge im ersten Obergeschoss zwischen ihm und Herrn Tomschke, den er schließlich mit einem Lederriemen in seinem Schlafzimmer zu Tode strangulierte.

Anschließend warf er die Leichen in den Hausbrunnen im Keller des Hauses zusammen mit diversen Textilien und einem Eimer, in dem sich aufgewischtes Blut, ein Putzlappen und ein Teil der Schädeldecke der Frau Tomschke befand.

Im Haus wurden mehrere Fußspuren, Fingerabdruckspuren und Blutspuren gefunden.

4 Analyse verschiedener Add-ons

4.1 Tatorterstellung

Bei der Tatorterstellung in Blender sind mehrere Aspekte wichtig. Die Zielgruppe, die mit dem Programm später arbeiten soll, sind diejenigen, die sich mit der Tatortrekonstruktion beschäftigen. Es ist davon auszugehen, dass diese kaum bis gar keine Praxis im 3D-Modelling haben. Insbesondere Blender ist oft nicht die erste Wahl als 3D-Programm, da es weniger bekannt als 3Ds-Max oder Maya ist. Zwar können diese Programme in Bezug auf das 3D-Modelling in etwa das Gleiche, jedoch unterscheiden sie sich in vielerlei Hinsicht im Handling [Ca15]. So ist das Interface von 3Ds-Max und Maya sehr ähnlich, da beide von Autodesk entwickelt wurden, und konzentrierten sich im Layout insbesondere auf verschiedene Buttons, wo hingegen Blender sich eher auf Shortcuts konzentriert, und somit anfänglich für neue Benutzer sehr unübersichtlich und abschreckend wirken kann.

Auch würde die Wahl nicht bevorzugt auf Blender zur Tatortrekonstruktion fallen, wenn davon auszugehen ist, dass die Ermittler zuerst eine Schulung nehmen müssten, um mit dem Programm einwandfrei umgehen zu können.

Ein weiterer wichtiger Aspekt ist, dass die traditionelle Modellierung, d. h. die Modellierung aus primitiven Objekte, wie einem Würfel oder einer Kugel, nicht nur Erfahrung benötigt, sondern auch mit einem hohen Zeitaufwand verbunden ist, um ein repräsentatives Ergebnis zu erstellen. Dies gilt insbesondere dann, wenn eine maßstabsgetreue Reproduktion eines Objektes Ziel ist.

Aus dieser Problematik schließen sich mehrere wichtige Eigenschaften, auf die die Add-ons geprüft werden müssen. Um eine schnelle Modellierung des Tatortes zu gewährleisten, muss es möglich sein, z. B. eine Wohnung weitestgehend automatisiert zu erstellen. Weitergehend ist es wichtig, die Möglichkeit zu besitzen, genaue Maße einzugeben, damit der Tatort maßstabsgetreu wiedergegeben werden kann.

Damit das Programm auch für 3D-Anfänger geeignet ist, sollte möglichst keine händische Modellierung in diesem Prozess mit inbegriffen sein. Das ideale Add-on sollte aus einem übersichtlichen Interface mit einer automatisierten Erstellung von maßstabsgetreuen Gebäuden ausgestattet sein.

Auf diese Eigenschaften sollen zwei verschiedene Softwarepakete geprüft werden. Einerseits das Add-on „Archimesh“ von Antonio Vazquez [URL-6] und andererseits das Blender Add-in „Fluid Designer“ von Microvellum [URL-7].

Archimesh

Archimesh ist ein Add-on, das für die Erstellung von architektonischen Modellen entwickelt wurde. Der Nutzer hat mehrere Möglichkeiten einen Raum zu erstellen. Hierbei geht das Add-on insbesondere auf die schnelle und benutzerfreundliche Erstellung von Wänden und den dazugehörigen Objekten wie Türen und Fenstern ein. Bevor es benutzt werden kann, muss es aber von der entsprechenden GitHub-Seite [URL-8] heruntergeladen und anschließend installiert werden. Sobald das Add-on Archimesh heruntergeladen wurde muss wie folgt in Blender navigiert werden: File >> User Preferences >> Add-ons >> Install from file. Nun wird das heruntergeladenen Add-on ausgewählt und bestätigt, um das Add-on zu installieren.

Nachdem Archimesh installiert wurde ändert sich das Interface Blenders, und es stehen einem neue Möglichkeiten zur Erstellung von architektonischen Modellen zur Verfügung.

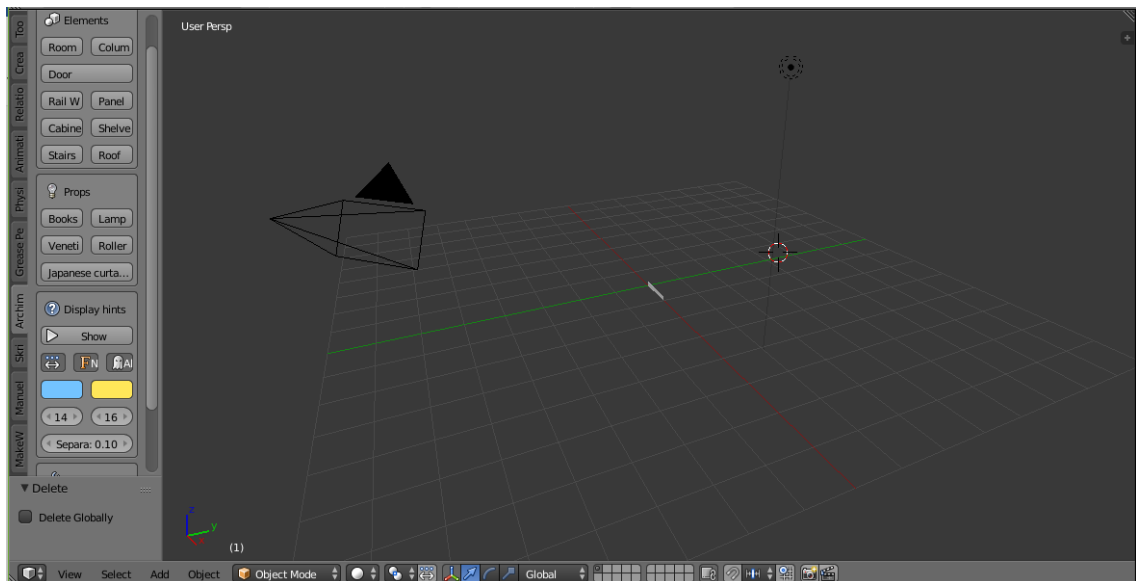


Abbildung 2: Blenderinterface mit dem neuen Reiter des Add-ons Archimesh. Auf der rechten Seite sind mehrere Buttons zu sehen, die es ermöglichen, verschiedene Objekte wie Räume, Türen, Fenster und Ähnliches zu erstellen.

Auf der linken Seite (siehe Abb. 2) des Blender-Interfaces gibt es nun einen neuen Reiter namens „Archimesh“. Hier sind verschiedene Buttons zu finden, die unterschiedliche Funktionen haben, um den Nutzer bei der Erstellung von Gebäuden zu unterstützen. Im Kontext der Tatorterstellung ist es weniger wichtig, die Gebäude äußerlich akkurat zu kreieren, vielmehr geht es um den maßstabsgetreuen Aufbau aller relevanten Räume und Objekte des Tatortes. So ist es beispielsweise nicht von Nöten, bei einem alten Bauwerk architektonische Details wie Säulen oder aufwändige Dächer, exakt wiederzugeben, sondern es sollte sich auf die relevanten Merkmale konzentrieren

werden. Solche Details können entweder abstrahiert wiedergegeben, oder aber diese bei der Erstellung des Tatortes weggelassen werden. Aus diesem Grund wird nicht weiter auf derartige Funktionen des Add-ons eingegangen, sondern ausführlich auf die relevanten, die zur Anfertigung des Tatortes brauchbar sind.

Ein Raum in Archimesh wird durch einen Klick auf den Button mit der Beschriftung „Room“, der sich im Kontextmenü „Elements“ befindet, realisiert. Dadurch öffnet sich ein neues Kontextmenü auf der rechten Seite des Interfaces.

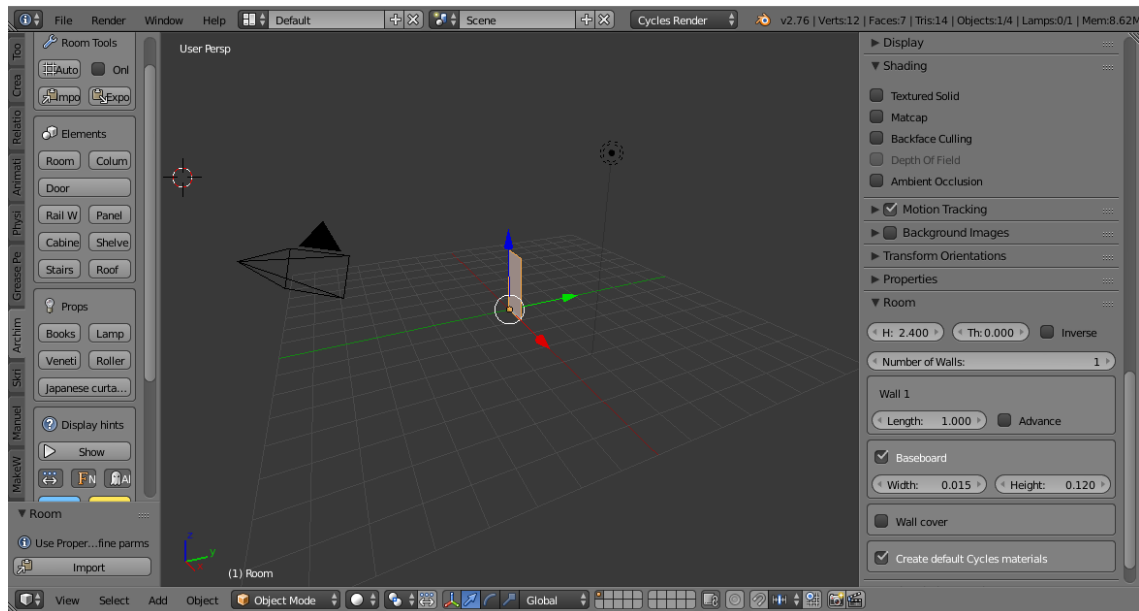


Abbildung 3: Interface, nachdem auf den Button „Room“ geklickt wurde. Im 3D-Viewport befindet sich nun ein neues Objekt. Dieses Objekt ist die erste Wand eines neuen Raumes, der mittels verschiedener Modifikationsmöglichkeiten auf der rechten Seite der Abbildung angepasst und erstellt werden kann.

Nun ist ein neues Objekt in der Mitte des Viewportes erschienen, das mit dem Kontextmenü auf der rechten Seite (durch den Shortcut „n“ wird dieses geöffnet und geschlossen) manipuliert werden kann. Beispielsweise kann die Anzahl der Wände des Raumes, der erstellt werden soll, durch klicken auf den Pfeil in der Schaltfläche „Number of Walls“ eingestellt werden. Die Schaltflächen „Height“ und „Thickness“ beeinflussen die Höhe und Dicke der Wände. Jede Wand wird einzeln in dem Menü auf der rechten Seite aufgeführt und kann in ihrer Länge beeinflusst werden. Alle Längeneinheiten beziehen sich auf eine Blendereinheit, die standardmäßig einem Meter entspricht.

Wie in Abb. 4 zu sehen ist, ist es zwangsweise nötig, die Länge einer Wand negativ anzugeben. Dies kommt dadurch zustande, dass die Wände mit einer Länge von +1 in eine Richtung von der Kante der vorherigen Wand zeigen, sobald eine neue Wand hinzugefügt wird. Wenn jedoch die Wand entlang der entgegengesetzten Richtung laufen soll, so muss die Länge der Wand negativ angegeben werden. Dies beeinflusst

die Länge der Wand jedoch nicht. Wird ein Haken bei „Advance“ gesetzt, ist es möglich, die Wände zu drehen, indem ein Haken bei „Angle“ gesetzt wird. Und sie, durch einen Haken bei „Curved“, zu verbiegen.

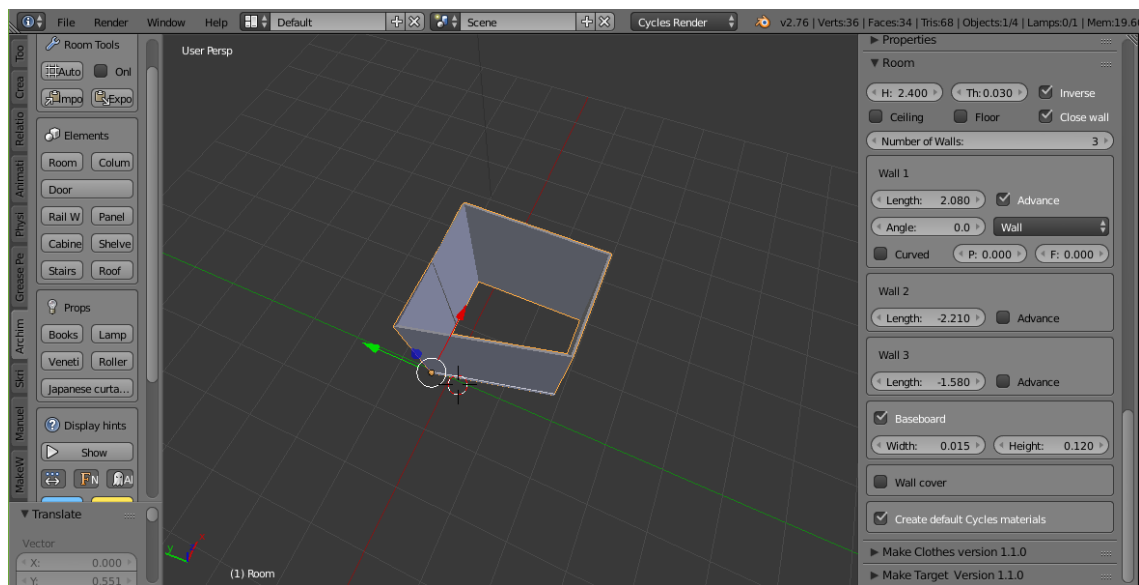


Abbildung 4: Erstellung eines Raumes mittels Archimesh. Die negativen Längen der Wände entstehen durch die Orientierung im 3D-Raum.

Außerdem ist es möglich, eine Fußleiste zu erstellen. Auch hier kann die Höhe und Breite der Leiste eingestellt werden, wenn ein Haken bei „Baseboard“ gesetzt wird. Wird ein Haken bei „Inverse“ gesetzt, so werden die Wände nach außen gedreht, dadurch wird die Fußleiste außen an der Wand platziert (siehe Abb. 4). Soll ein Fußboden oder eine Decke hinzugefügt werden, kann dies mittels Platzierung eines Hakens bei „Ceiling“, beziehungsweise „Floor“, realisiert werden. Mit dem Setzen eines Hakens bei „Close wall“ wird der Raum automatisch durch eine vierte Wand erweitert, die den Raum schließt. Diese kann aber nicht weiter modifiziert werden.

Nachdem ein Raum erstellt wurde, müssen noch Fenster und Türen eingefügt werden. Auch dies ist mit Archimesh lösbar. Um eine Tür zu erstellen wird der Button „Door“ auf der linken Seite unter „Elements“ benutzt. Dadurch wird eine Tür erstellt, die, wie die Wände, nun auf der rechten Seite des Viewporters weiter modifiziert werden kann. Wie in Abb. 5 zu sehen ist, ist es möglich, die Tür in ihrer Höhe, Breite, Dicke und Größe zu beeinflussen. Des Weiteren kann ein anderes Türmodell ausgewählt werden, und auch die Seite, zu der die Tür öffnet bestimmt werden. Auch gibt einem Archimesh die Möglichkeit verschiedene Türknaufe zu wählen.

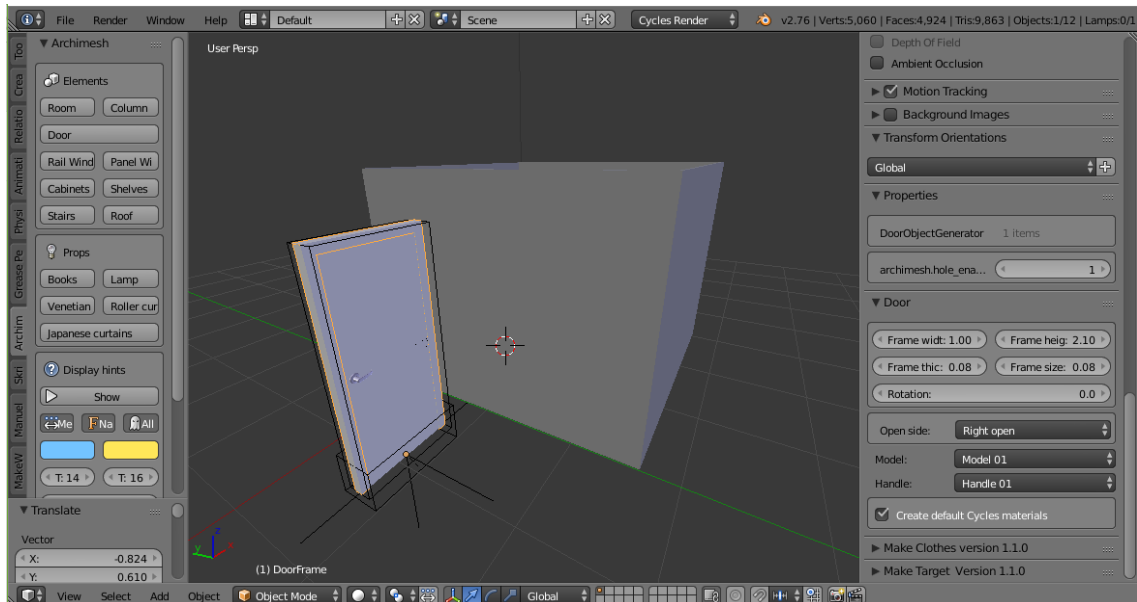


Abbildung 5: Blenderinterface nachdem eine Tür erstellt wurde. Auf der rechten Seite sind die verschiedenen Einstellungsmöglichkeiten zu sehen: Rahmenbreite, -höhe, -größe und -dicke. Um die Tür zu bewegen muss nicht die Tür, sondern das Kreuz aus dünnen Linien unterhalb der Tür ausgewählt werden.

Um die Tür in der Wand an der passenden Stelle zu platzieren, muss das Kreuz unter der Tür ausgewählt werden (siehe Abb. 5), nun kann die Tür rotiert, skaliert und verschoben werden. Ist die Tür an der richtigen Stelle in der Wand platziert, sollte nun die Wand angewählt werden und durch die „Autoholes“ Funktion ein Loch in die Wand geschnitten werden. Hierzu muss der Button namens „Autoholes“ unter der Kategorie „Room Tools“ gedrückt werden. Manchmal muss die Tür hin- und her bewegt werden, damit dies fehlerfrei funktioniert. Dieser Prozess kann für die Platzierung der Fenster analog angewandt werden.

Fluid Designer

Der Fluid-Designer von Microvellum ist ein Standalone für Windows, Mac OS und Linux und ist auf Blender basierend aufgebaut. Es handelt sich um eine modifizierte Version von Blender, weswegen es nicht mehr als Add-on angesehen werden kann und als eigenes Programm heruntergeladen und installiert werden muss.

Laut Microvellum ist Fluid-Designer ein Tool zur Erstellung von Innenarchitektonischen Modellen. [URL-9] Hierbei konzentriert sich das Programm insbesondere auf die Präsentation der Räume. Das Interface ist sehr übersichtlich gestaltet und lässt sich intuitiv bedienen.

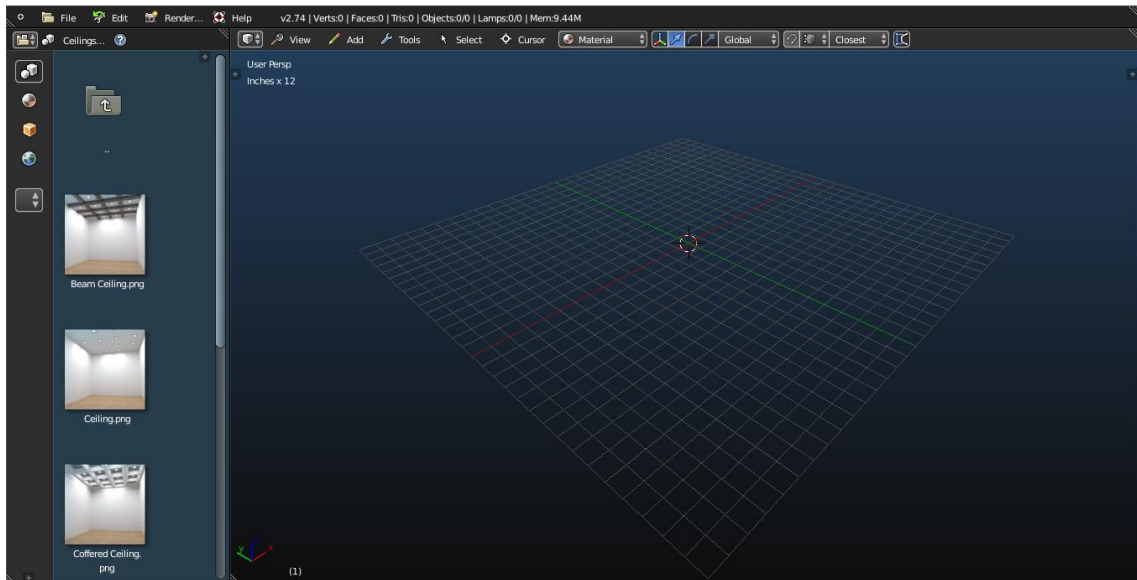


Abbildung 6: Interface des Fluid Designers - es ist ersichtlich, dass das Programm auf Blender aufbaut, aber dennoch wesentlich modifiziert wurde.

Auf der linken Seite des Interfaces gibt es mehrere Icons, die unterschiedliche Funktionen aufweisen. Um einen Raum zu erstellen wird der Shortcut „Shift + A“ verwendet, der im Generellen in Blender zur Erstellung neuer Objekte gebräuchlich ist.

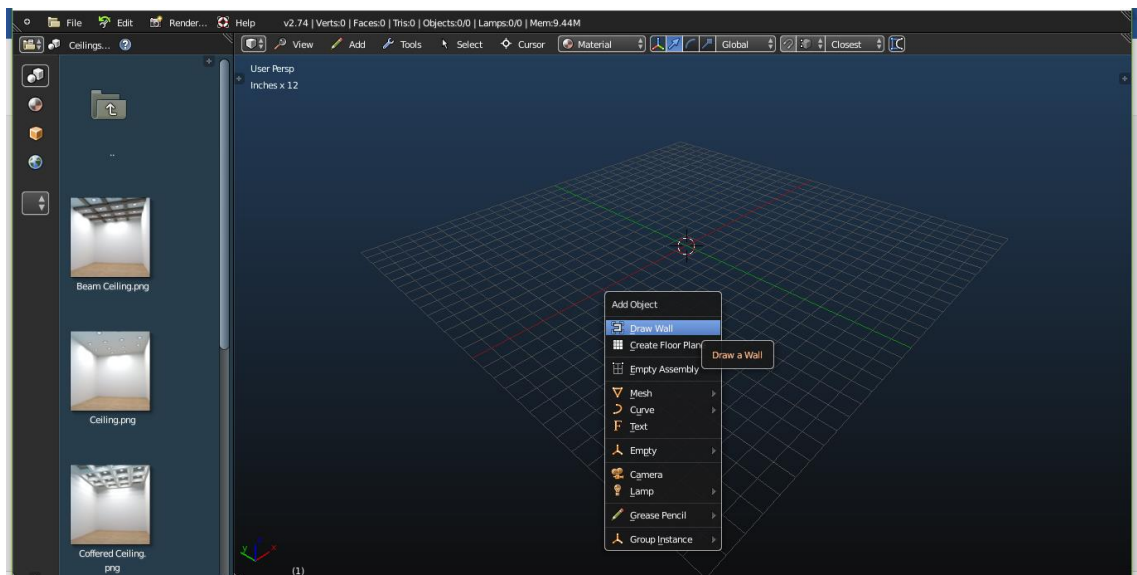


Abbildung 7: Eine neue Wand wird durch "Draw Wall" erstellt. Das Kontextmenü wird durch Shift + A geöffnet.

Nachdem eine Wand erstellt wurde, ist es möglich, diese in ihren Dimensionen anzupassen. Jedoch sollte zuvor das metrische System (bei Fluid-Designer ist dies standardmäßig in Millimetern angegeben) oder keine Einheiten eingestellt werden, da der

Fluid-Designer bei der Erstöffnung mit Inches rechnet. Hierzu wird folgendermaßen navigiert: File >> Units >> Unit System >> Metric /None.

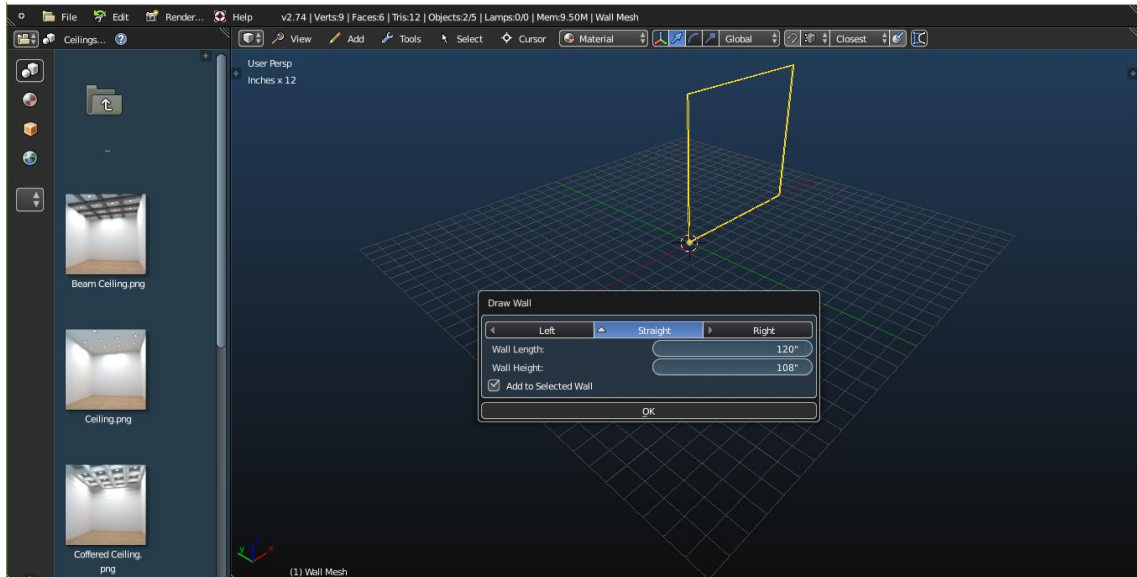


Abbildung 8: Erstellung einer Wand in Fluid Designer - Es ist möglich, Ausrichtung, Länge und Höhe der Wand festzulegen.

Nun ermöglicht das Pop-up-Menü erste Einstellungen an der Wand vorzunehmen. So kann sie nach rechts, links oder gerade ausgerichtet werden, aber auch die Länge und Höhe können angegeben werden. Nachdem der Dialog bestätigt wurde, kann durch einen Rechtsklick auf die Wand auch die Dicke, Neigung und Platzierung der Wand eingestellt werden. Um eine zusätzliche Wand zu erzeugen muss wieder der Shortcut „Shift + A“ verwendet werden. Auch diese hat die gleichen Einstellungsmöglichkeiten wie in Abb. 8. Jedoch ist hier nun ersichtlich, weshalb das Pop-up-Menü die Möglichkeit gibt, die Wand rechts, links oder gerade auszurichten. Diese bezieht sich auf die Orientierung der vorigen Wand. So kann die neue Wand entweder von der letzten Wand aus nach rechts, links oder fortlaufend anschließen. Zu beachten ist, dass bevor eine neue Wand erstellt wird, zuerst diejenige Wand ausgewählt (durch einen Linksklick auf die entsprechende Wand) werden sollte, an die sich die neue anschließen soll. Andernfalls kann es dazu führen, dass ein neuer Raum erstellt wird. Ist der Raum erstellt, kann abermals durch den Shortcut „Shift + A“ ein Fußboden erstellt werden. Diese Funktion heißt „Create Floor Plane“.

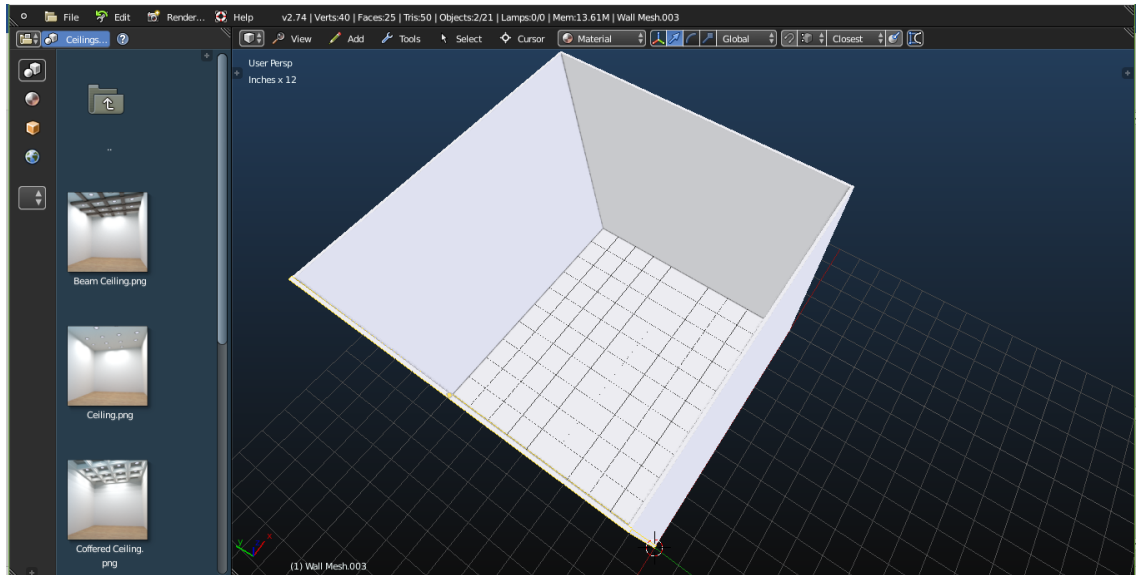


Abbildung 9: Ein erstellter Raum mit einem Fußboden

Um den Raum mit Türen und Fenstern zu vervollständigen, können voreingestellte Türen und Fenster verwendet werden. Hierzu wird auf der linken Seite des Interfaces neben dem Ordner-Icon auf die Schaltfläche gedrückt, die beim ersten Öffnen „Ceilings“ heißt (siehe Abb. 9). Nun können mehrere Objektgruppen gewählt werden, wie beispielsweise „Doors“ für Türen oder „Windows“ für Fenster. Nachdem ein Modell ausgewählt wurde, kann dieses nun per Drag-And-Drop in eine Wand eingefügt werden.

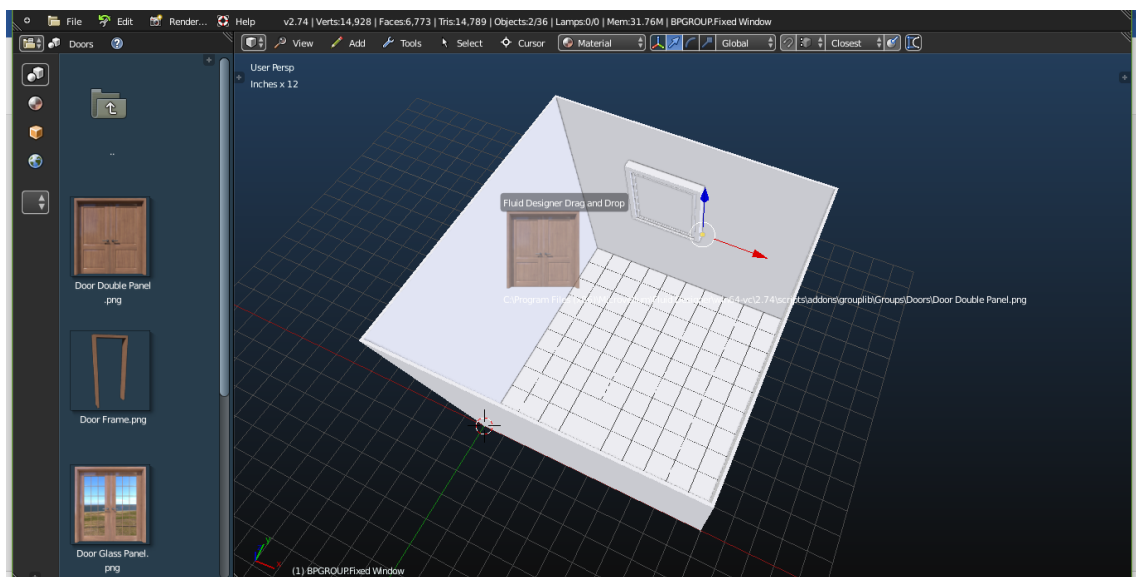


Abbildung 10: Nachdem ein Modell auf der rechten Seite ausgewählt wurde, kann es per Drag and Drop auf die entsprechende Wand gezogen werden.

Ein starkes Tool des Fluid Designers ist die mitgelieferte Objektbibliothek. Leider ist die aktuellste Version für Windows (2015 R2) fehlerhaft. Deswegen ist es nicht möglich, die Objektbibliothek vollständig zu benutzen.

4.2 Personenerstellung

Bei der Personenerstellung sind ähnliche Aspekte, wie bei der Tatorterstellung, wichtig. Auch hier soll dem Benutzer der Software kaum eigene Modellierung zugemutet werden. Des Weiteren ist ein übersichtliches Interface und eine schnelle, sowie genaue Erstellung des gewünschten Ergebnisses ein wichtiges Element zur Auswahl des geeigneten Softwarepaketes.

Bei der Personenerstellung sind mehrere Faktoren, insbesondere bei der Tätererstellung, wichtig. So muss die Software imstande sein, Körpergröße, Ethnie und Körperform, ergo die augenmerkliche Fitness der Person, sowie das Geschlecht und Alter wieder zu geben.

Hierzu werden ebenfalls, wie bei der Tatorterstellung, zwei Softwarepakete miteinander verglichen und auf ihren Nutzen geprüft. Einerseits das bereits erwähnte Add-on Manuel-Bastioni-Lab von Manuel Bastioni, andererseits das Standalone makeHuman von dem makeHuman Team. Beide sind sehr ähnlich im Umfang der Software, da Manuel Bastioni ebenfalls ein Mitglied des makeHuman Teams ist, und an dem Standalone makeHuman mitgewirkt hat. Jedoch wurde nicht jede Funktion des Programmes in Blender von makeHuman übernommen.

MakeHuman

Das Programm konzentriert sich auf die Manipulation der Körper durch verschiedene Schieberegler (siehe Abb. 11). Unter dem Tab „Modelling“ sind einige Einstellungsmöglichkeiten zu finden. So ist es möglich unter dem Untertab „Main“, erste grundlegende Einstellungen des Charakters vorzunehmen. Beispielsweise kann das Geschlecht, das Alter, die Größe und das Gewicht verändert werden. Auch ist es möglich verschiedenen Ethnien wie kaukasisch, asiatisch oder afrikanisch anteilig einzustellen.



Abbildung 11: Interface des Programms MakeHuman - Alle Einstellungen können durch Slider vorgenommen werden. Am unteren Bildschirmrand sind die genauen Werte zu sehen.

MakeHuman kommt mit einigen Funktionen zur Modellierung mit sich. So ist es möglich, geschlechterspezifische Merkmale detailliert zu formen, aber auch das Gesicht kann genau angepasst werden. Auch der Torso, Arme und Beine können nach Belieben verändert werden. Ist es nötig, genaue Maße zu verwenden, so kann dies unter dem Untertab „Measure“ realisiert werden.



Abbildung 12: Es ist möglich, das Gesicht detailreich zu modeln. Auch hier wird auf Slider gesetzt.

Unter dem Tab „Geometries“ können Kleidung, Augen, Zähne und Augenbrauen eingestellt werden. Auch Haare in verschiedenen Stilen und Farben können hier ausgewählt und angewendet werden.

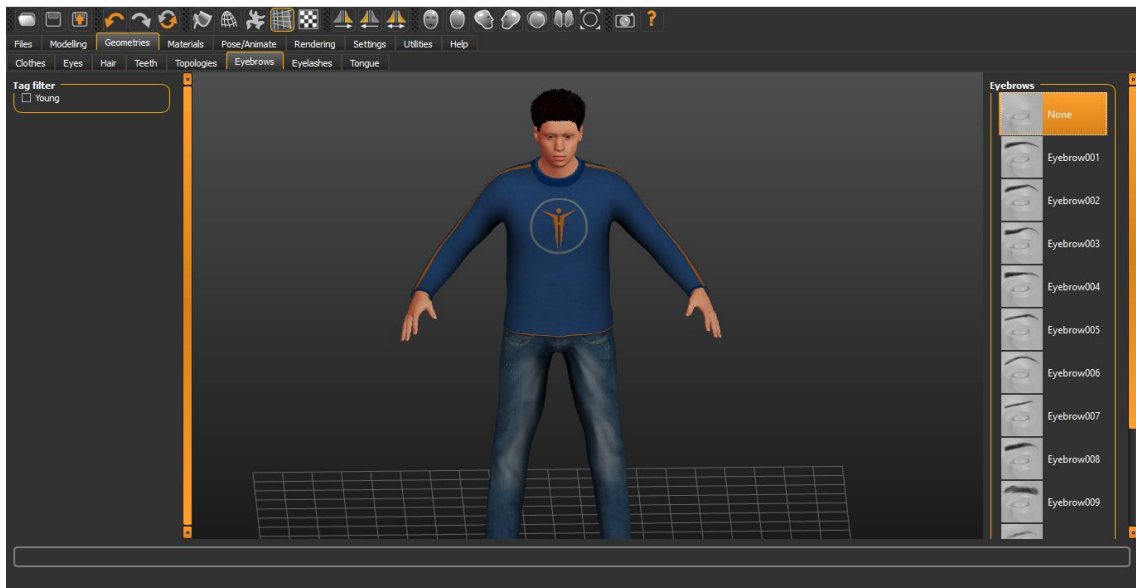


Abbildung 13: Ein Charakter der mit Kleidung und Haaren versehen wurde.

Abschließend kann der Charakter noch mit Materialien versehen werden. Hierzu kann unter dem Tab „Materials“ die Hautfarbe, Haarfarbe und Augenfarbe eingestellt werden. Soll der Charakter anschließend noch animiert werden, muss unter „Animate/Pose“ beim Untertab „Skeleton“ ein Skelet ausgewählt werden. Dies kann z.B. das „Mocap-Skeleton“ sein, wenn später mit Motion-Capture-Dateien der Charakter bewegt werden soll. Unter dem Tab „Files“ ist es abschließend möglich, den Charakter zu speichern und in andere Formate zu exportieren. Hier sollte darauf geachtet werden, dass, wenn eine Weiterverarbeitung in Blender erwünscht ist, nicht das Object-Format gewählt wird, da dieses keine Weitergabe der Materialien und Texturen ermöglicht.

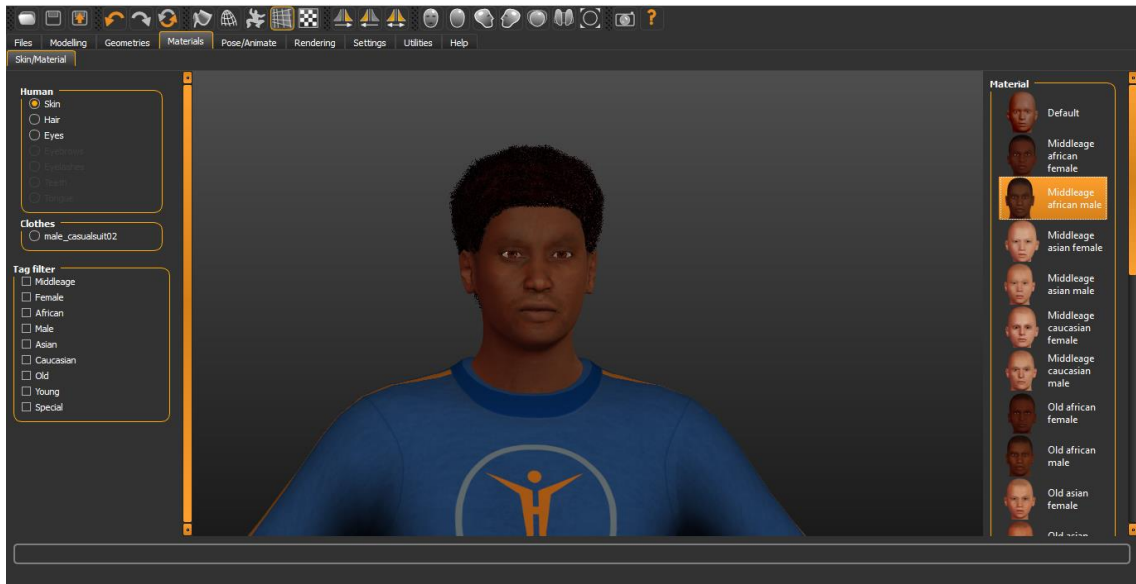


Abbildung 14: Unter dem Tab "Materials" können Hautfarbe, Haarfarbe und Augenfarbe eingestellt werden. Hierzu werden Voreinstellungen verwendet, die nach Ethnie und Alter sortiert sind.

Manuel-Bastioni-Lab

„Manuel Bastioni Laboratory“ oder auch Manuel-Bastioni-Lab ist ein umfangreiches Add-on für Blender. Dieses muss, wie schon zuvor in 4.1 beschrieben, installiert werden. Das Layout unterscheidet sich von makeHuman grundlegend. Wie in Abb. 15 zu sehen ist, ist das Add-on nun ein neuer Tab auf der linken Seite des Interfaces.

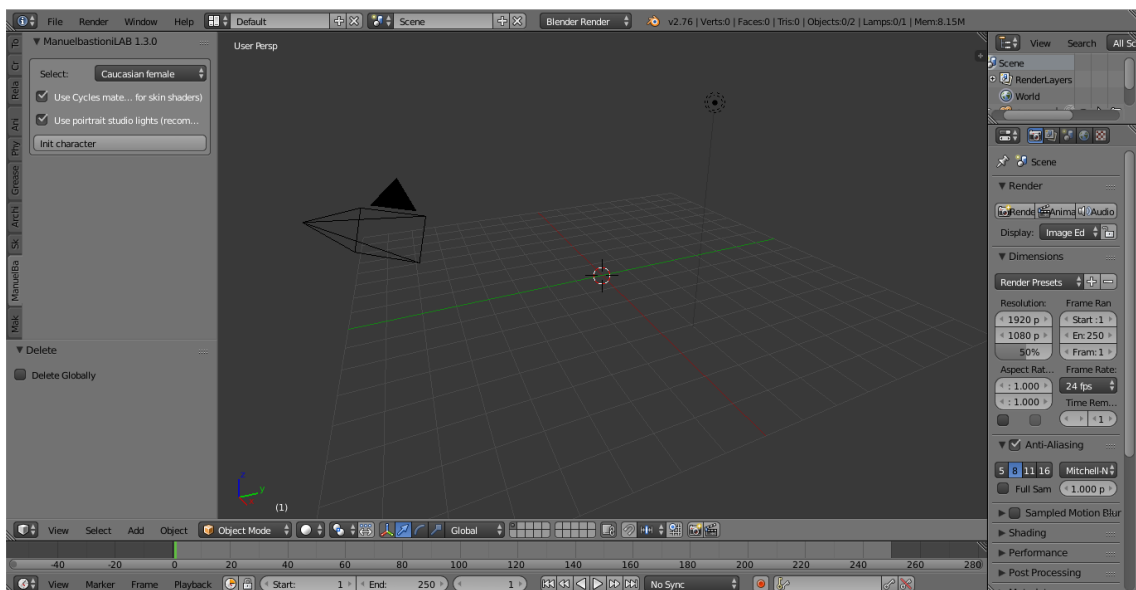


Abbildung 15: Das Add-on benutzt die Cycles-Render-Engine von Blender, um realistische Hauttexturen zu verwenden. Dies ist aber optional verwendbar. Um einen Charakter zu erstellen, muss „Init character“ gedrückt werden.

Um einen Charakter zu erstellen, muss zwischen kaukasisch, asiatisch und afrikanisch, sowie zwischen männlich und weiblich entschieden werden. Mit einem Klick auf „init. character“ kann nun der Charakter erstellt werden.

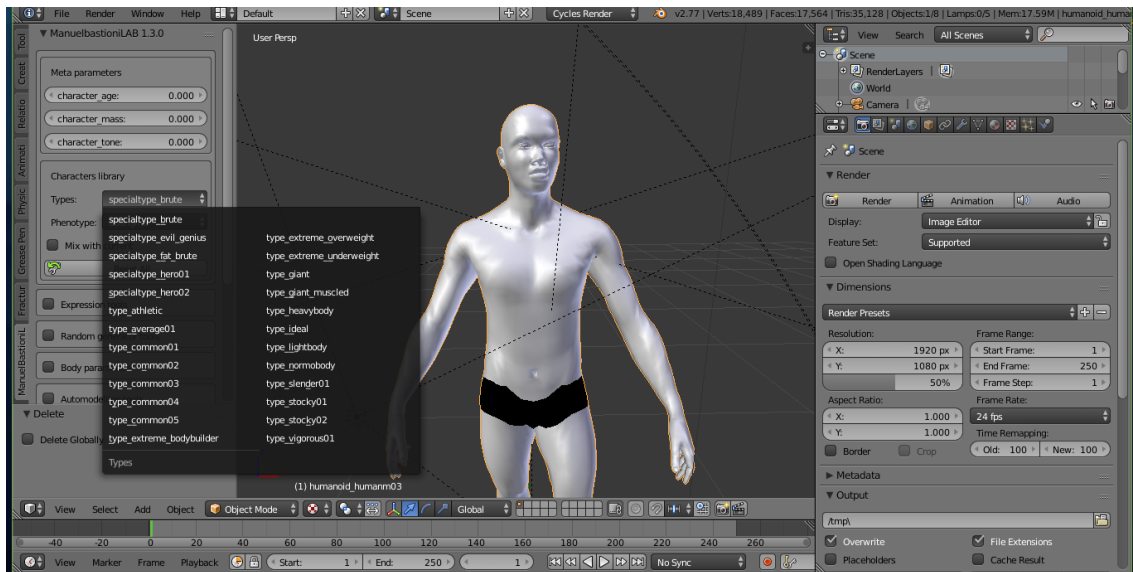


Abbildung 16: Nachdem der Charakter erstellt wurde, kann er durch verschiedene Voreinstellungen angepasst werden.

Nun kann unter „Types“ ein bestimmter Typus eingestellt werden. Beispielsweise kann der Körper sehr trainiert sein, aber auch extrem über- oder untergewichtig. Auch die Merkmale verschiedener Herkunftsregionen können unter „Phenotype“ verfeinert werden. So kann, wenn zuvor die afrikanische Abstammung gewählt wurde, nun zwischen „aboriginal“, „african“, „afroasian“ und „afromediterranean“ entschieden werden. Zusätzlich kann unter „Meta parameters“ das Alter, das Gewicht und die Muskelmasse bestimmt werden. Hierbei reichen die Werte von -1,000 bis +1,000.

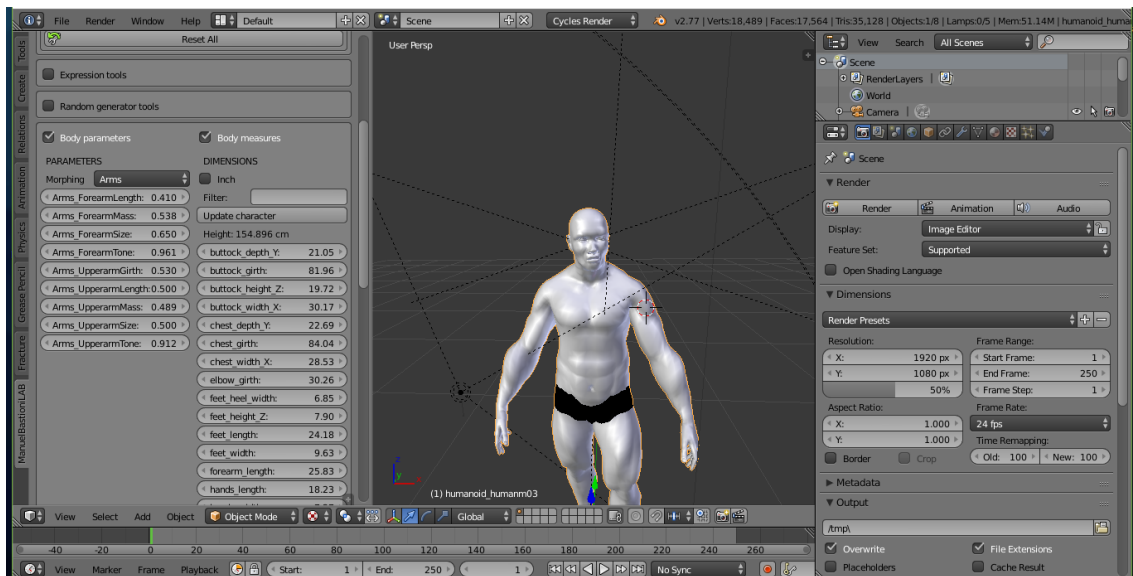


Abbildung 17: Jeder Körperteil kann individuell mittels Schieberegler angepasst werden. Wird nach einer Veränderung der Werte „Update character“ gedrückt, berechnet das Add-on die Werte so, dass es realistisch aussieht.

Um nun den Körper weitergehend anzupassen, muss ein Haken bei „Bodyparameters“ gesetzt werden. Nun kann ein Körperteil via Drop-Down-Menü (siehe Abb. 17) ausgewählt werden, und dieses mittels Ziehen der Werte in eine Richtung (ähnlich wie ein Schieberegler) oder die genaue Eingabe der Werte verändert werden. Wurden alle Werte eingegeben, ist es möglich durch „Update character“ die Massenverteilung des Körpers neu zu berechnen. Damit ist eine möglichst reale Verteilung von Fett und Muskeln gewährleistet.

Nachdem der Charakter grundlegend erstellt wurde, können nun die Haut und deren Details angepasst werden. Hierzu wird der Skin-Editor benutzt (siehe Abb. 18). Wird hier „Enable subdivision preview“ und „Enable displacement preview“ ausgewählt, werden Muskeln und Adern realistisch, je nach Muskel- und Fettverteilung, angezeigt.

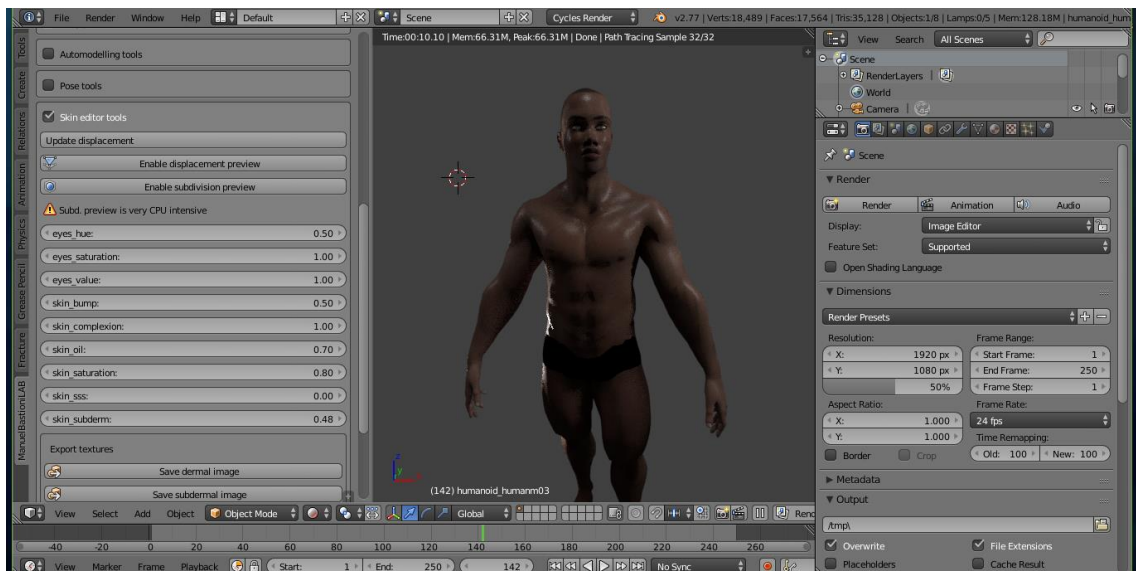


Abbildung 18: Durch den Skin-Editor kann die Haut des Körpers realistisch dargestellt werden. Hier können Farbe, Oberflächendetails und Beschaffenheit der Haut eingestellt werden.

Mit diesem Modell kann nun weiter in Blender gearbeitet werden. Bei der Initialisierung des Charakters wurde ein Skelet für ihn erstellt; So kann er für Animationen weiterverwendet werden. Hierzu muss auf „Finalize“ gedrückt werden. Danach kann der Charakter nicht mehr durch das Add-on verändert werden.

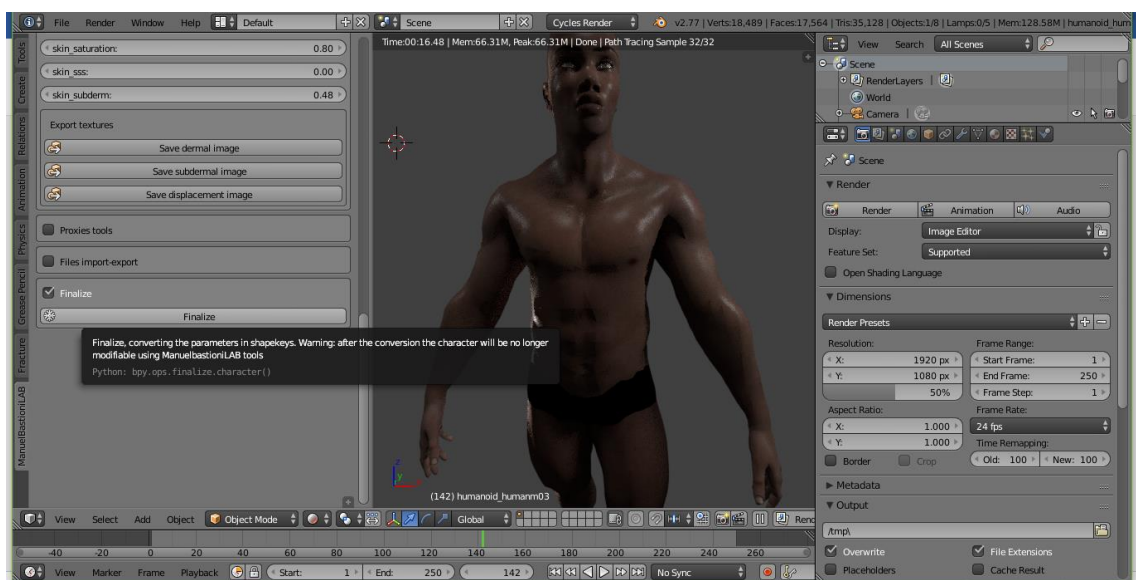


Abbildung 19: Durch das Finalisieren wird der Charakter in ein Blender-Objekt umgewandelt und kann als solches weiterverwendet werden.

4.3 Visualisierung von Bewegungsmustern

Bei der Visualisierung von Bewegungsmustern wird auf zwei unterschiedliche Punkte eingegangen. Zum einen auf die Visualisierung von Fäulnisgasen, da in manchen Fällen diese ausschlaggebend sind, eine Leiche zu finden. Zum anderen auf die Darstellung der Bewegungsabläufe von Täter und Opfer zu Zeiten der Tat.

Im ersten Teil soll auf die Bewegungsmuster der involvierten Personen eingegangen werden. Hierzu wird das vom makeHuman Team bereitgestellte Add-on „makeWalk“ benutzt und auf seine Zweckdienlichkeit geprüft.

Im zweiten Teil des Teilkapitels soll auf die Demonstration der Fäulnisgase eingegangen werden. Hierzu wird das blenderinterne Partikelsystem und das Add-on „CubeSurfer“ von Jean-Francois Gallant [URL-10] betrachtet.

MakeWalk

Das MakeWalk-Add-on ist zur einfachen Animierung der Skelete (oder auch „Rigs“) eines Charakters erstellt worden. Um den Charakter zu animieren, wird dafür auf Motion-Capture-Dateien zurückgegriffen, die dann auf das Rig des Charakters übertragen werden. Da es einige kostenlose Datenbanken mit Motion-Capture-Dateien gibt, ist es so möglich, nahezu jeden Bewegungsablauf auf ein Rig zu übertragen. Ein Beispiel für eine kostenlose Datenbank wäre die Plattform „CgSpeed“ [URL-11].

Um das Addon MakeWalk zu installieren, muss es erst von der Website des MakeHuman-Teams heruntergeladen werden. Die erhaltene Datei sollte nun extrahiert werden und muss danach in den Add-on Ordner Blenders verschoben werden. Dieser ist unter Windows 10 unter folgendem Dateipfad zu finden: C:\Users\Username\AppData\Roaming\Blender Foundation\Blender\2.76\scripts\addons. Wird nun Blender neu geöffnet, sollte das Add-on als neuer Tab in der Tool-Bar gefunden werden.



Abbildung 20: Der Charakter aus dem MakeHuman Standalone wurde in Blender importiert. Damit er animiert werden kann, muss sein Rig ausgewählt werden. Auf der rechten Seite im Outliner kann das Skelett des Charakters einfach ausgewählt werden. Nun können alle Funktionen des Add-ons benutzt werden.

Um das Rig, und somit den Charakter, zu animieren, muss zuerst das Rig ausgewählt werden. Dies kann im Outliner (siehe Abb. 20) gefunden und mit einem Klick darauf angewählt werden. Dieses Rig sollte das Motion-Capture-Rig aus dem MakeHuman Standalone, das Standard-Rig aus dem Manuel Bastioni Lab oder das Blender Rig „Rigify“ sein. Nun muss auf „Load And Retarget“ geklickt werden, wodurch sich ein neues Fenster (siehe Abb. 21) öffnet. Hier muss das gewünschte Motion-Capture-File ausgewählt werden.

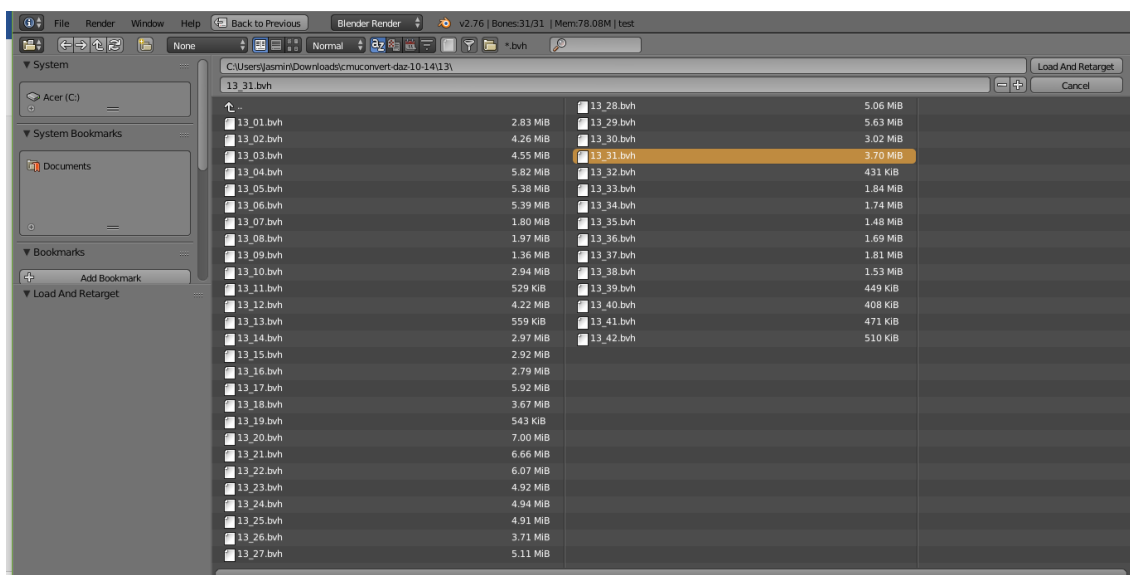


Abbildung 21: Wurde auf "Load And Retarget" gedrückt, wird zu dem Ordner in dem sich die Motion-Capture-Dateien befinden navigiert und die gewünschte ausgewählt.

Nun kann die Animation abgespielt werden. Soll eine andere Motion-Capture-Datei benutzt werden, so kann der Vorgang beliebig wiederholt werden.

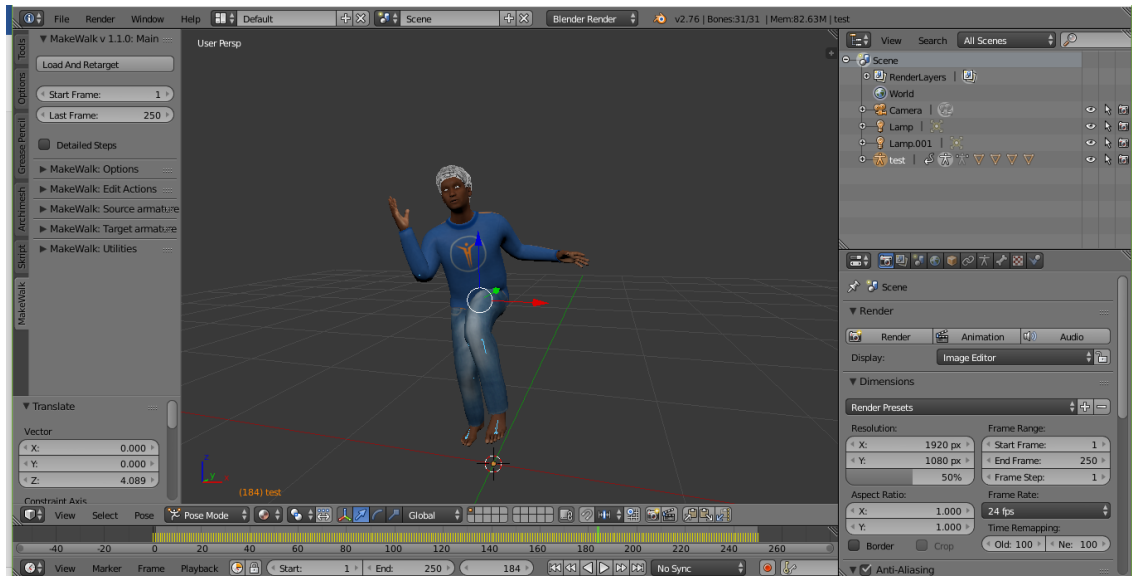


Abbildung 22: MakeWalk übersetzt die Motion-Capture-Datei in einzelne Keyframes. So kann nun die Animation abgespielt werden.

Blenders Partikelsystem

Das Partikelsystem von Blender ist sehr umfangreich und kann für einige Operationen verwendet werden. Wird ein Partikelsystem zu einem Körper hinzugefügt, dient dieser Körper als Ursprungspunkt für die Partikel. Diese werden über eine gewisse Zeitspanne vom Körper aus erzeugt und können in ihrem Verhalten manipuliert werden. So ist es möglich, Rauch oder auch Wasser zu simulieren. In der Tatortrekonstruktion könnten Partikelsysteme aber auch zur Visualisierung von Bewegungsmustern, Fäulnisgasen oder der Verteilung von Blut benutzt werden.

Um ein Partikelsystem anzulegen, wird der Körper, der der Ursprung der Partikel werden soll, angewählt. Nun kann im Properties-Panel der Partikelsysteme-Tab ausgewählt (siehe Abb. 23), und ein neues Partikelsystem erstellt werden, indem auf das Plus gedrückt wird.

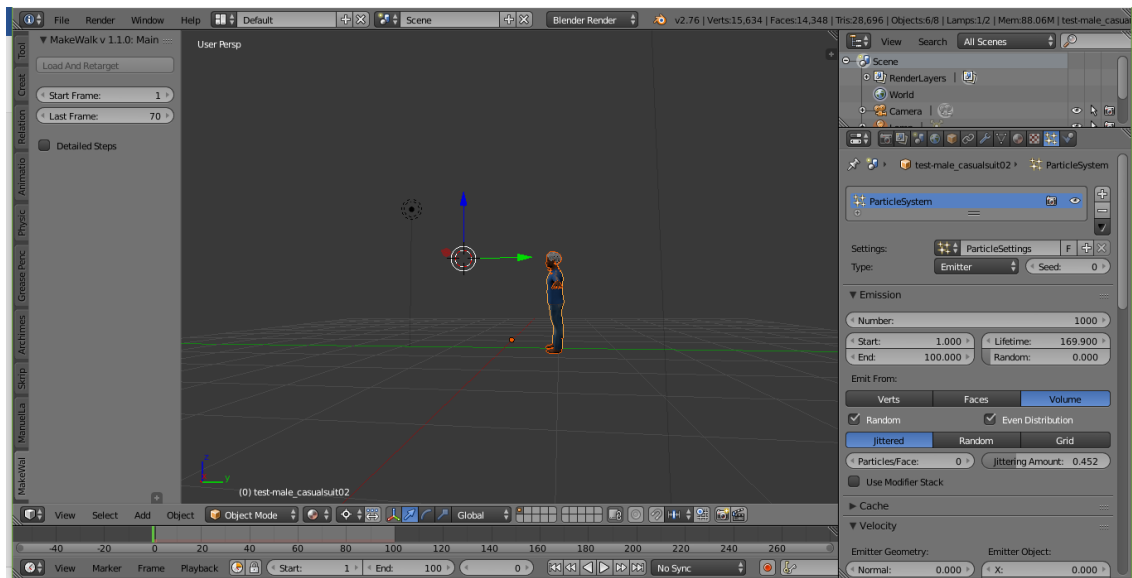


Abbildung 23: Unter dem Partikelsystem-Tab können neue Systeme angelegt werden. Diese können nun in ihrer Größe, Zeitspanne und vielem mehr verändert werden.

Um den Weg eines Menschen mittels Partikelsystem zu visualisieren, muss sich das Partikelsystem mit dem Menschen mitbewegen, und sollte dabei hinter der Person für einige Zeit sichtbar sein. Die Zahl der Partikel wird hierfür beispielsweise auf 1000 gesetzt, die Startzeit auf Beginn der Animation und die Lebensspanne (Lifetime), also die Anzeigedauer der Partikel, über 100 Frames. Dies bedeutet, dass 1000 Partikel ab Beginn der Animation erstellt werden, und diese über 100 Frames sichtbar sind. Damit die Partikel sich hinter der Person her bewegen, wird in den Physics-Eigenschaften „Boids“ ausgewählt und ein Haken bei „Allow Flight“ gesetzt. So können die Partikel in der gleichen Höhe, in der sie erstellt wurden, bleiben, und fallen nicht zu Boden. Nun können noch weitere Werte, je nach gewünschtem Effekt eingestellt werden, beispielsweise beeinflusst „Air Speed“ die Schnelligkeit der Partikel.

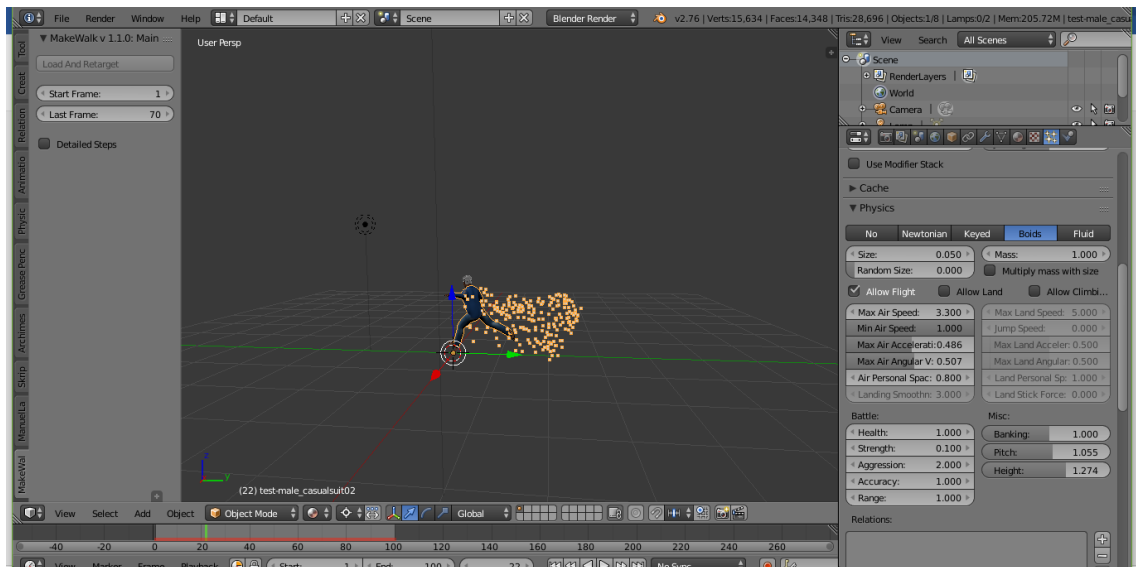


Abbildung 24: Durch das Partikelsystem ist es möglich, den gelaufenen Weg des Charakters zu visualisieren.

Sollen Fäulnisgase visualisiert werden, so müssen die Partikel vom Körper nach oben in die Luft erzeugt werden und sich kegelförmig ausbreiten. Dazu kann bei den Physics-Einstellungen „Newtonian“ ausgewählt werden und anschließend der Tangent-Wert unter „Velocity“ beispielsweise auf 1.15 gestellt werden. Damit die Partikel nicht nach unten fallen, sollte unter „Field Weights“ die Gravitation auf einen geringen negativen Wert wie -0.1 gestellt werden. Je niedriger dieser Wert ist, umso schneller bewegen sich die Partikel.

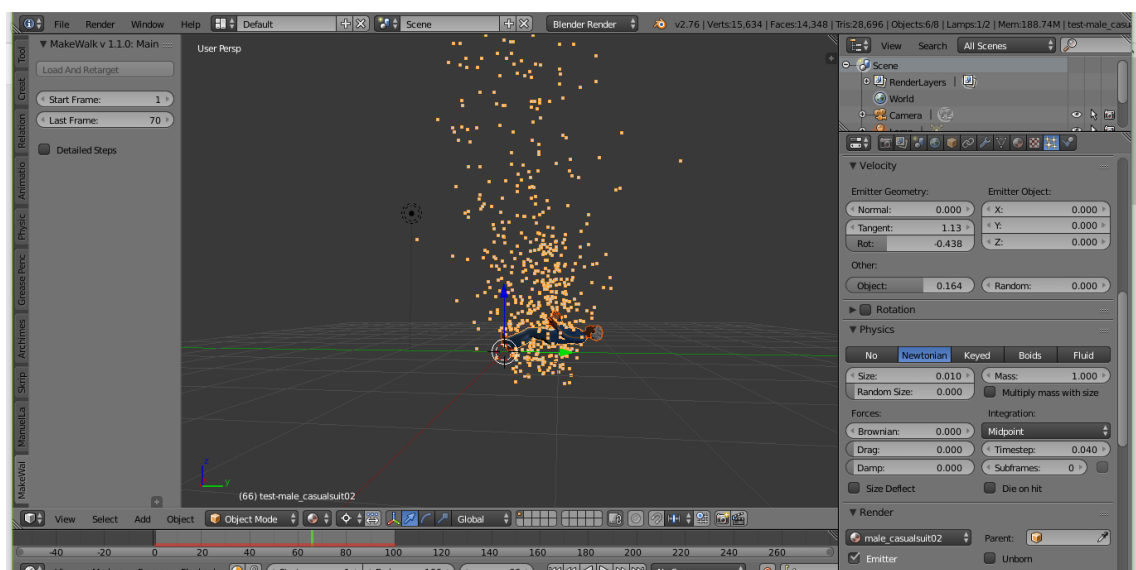


Abbildung 25: Indem die Gravitation auf -0.5 gesetzt wird, fliegen die Partikel nach oben und können so Gase simulieren. Um die Partikel kegelförmig vom Körper weg bewegen zu lassen, wird der Tangent-Wert erhöht.

CubeSurfer

Das CubeSurfer-Add-on ist eine Erweiterung des Partikelsystems. Es kann um jedes Partikel einen Körper erstellen, was in der praktischen Anwendung dazu führt, dass schnell Flüssigkeiten simuliert werden können.

Das Add-on wird, wie zuvor beschrieben, in Blender installiert. Um ein Partikelsystem, das sich wie eine Flüssigkeit verhält, zu erstellen muss, unter „Physics“ „Fluid“ eingestellt werden. Nun können die Werte je nach Flüssigkeit angepasst werden. Hierbei sind insbesondere die Werte für Viskosität und Steifheit ausschlaggebend.

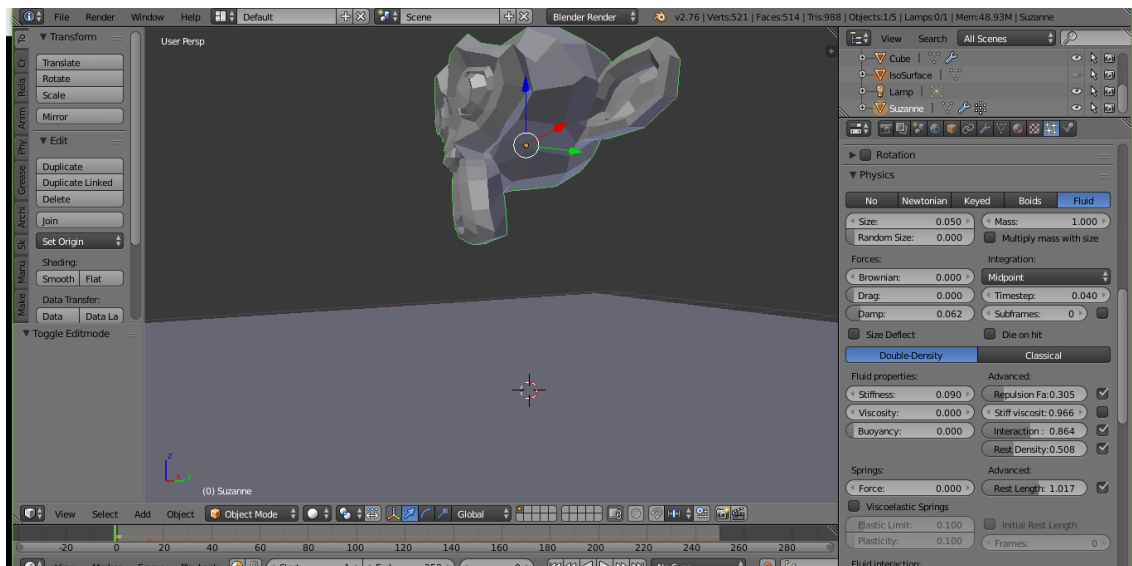


Abbildung 26: Das Partikelsystem wird mittels der Einstellung „Fluid“ an die Verhaltensweisen von Flüssigkeiten angepasst. Beispielsweise kann die Viskosität und Steifheit der Flüssigkeit eingestellt werden.

In diesem Beispiel wird der Kopf auf den Boden fallen gelassen. Beim Aufprall soll an der Aufprallstelle Blut austreten. Dazu wird dem Kopf ein Rigid-Body (siehe 4.4 zur Erklärung der Physics-Einstellungen in Blender) und dem Boden die Physics-Eigenschaft „Collider“ zugeteilt. Des Weiteren werden „Particle-Stickiness“ und „Particle-Damping“ höhergestellt, damit gewährleistet ist, dass die Partikel am Boden haften.

Damit die Partikel nur von der Aufprallstelle erzeugt werden, wird mittels Tab-Taste in den Edit-Modus gewechselt. Nun kann mit dem Shortcut „C“ ein Auswahlwerkzeug benutzt werden. Mit diesem wird die Aufprallstelle markiert. Nun wird eine Vertex-Gruppe im Properties-Panel unter dem Vertex-Group-Tab angelegt. Hierzu wird durch Klicken auf das Plus eine Vertex-Gruppe (siehe Abb. 27) erzeugt und anschließend beliebig benannt.

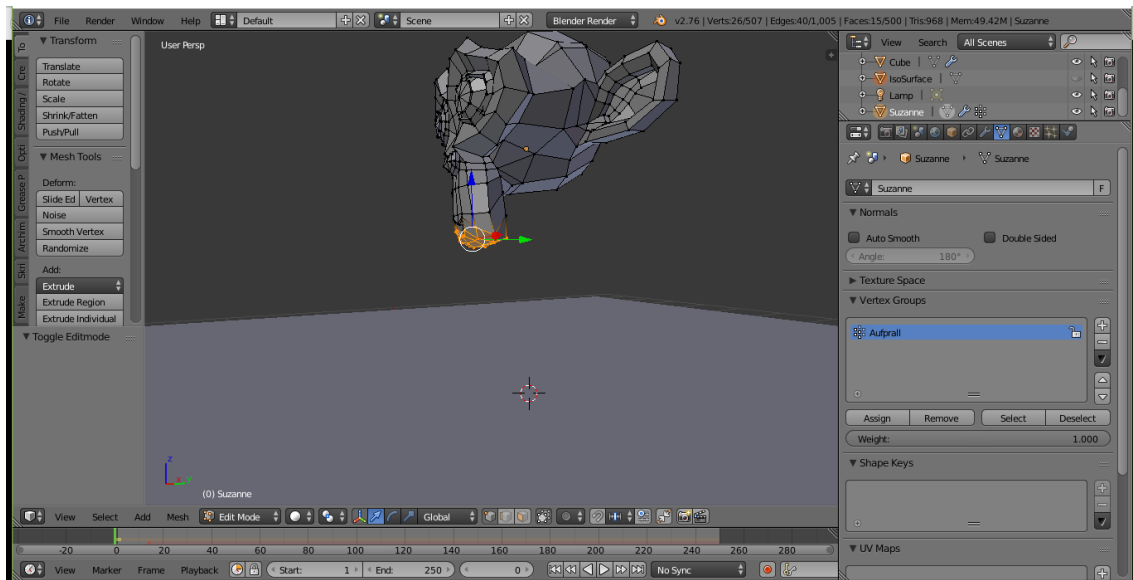


Abbildung 27: Die Aufprallstelle wird markiert und eine neue Vertex-Gruppe erstellt. Eine Vertex-Gruppe ist eine Auswahl verschiedener Mesh-Punkte eines Objektes. Diese Auswahl kann nun selektiert, deselektiert und für verschiedenen Anwendungen benutzt werden.

In den Partikeleinstellungen muss anschließend unter „Vertex-Groups“ bei „Density“ die soeben erstellte Vertex-Gruppe ausgewählt werden.

Damit erst beim Aufprall des Kopfes das Partikelsystem erzeugt wird, wird in den Einstellungen ebenfalls der Startpunkt des Partikelsystems auf jenen Frame gelegt, bei dem der Kopf auf den Boden trifft.

Nun kann das CubeSurfer Add-on benutzt werden, um aus den einzelnen Partikeln eine Flüssigkeit zu machen. Hierzu wird im Viewport „Shift + A“ gedrückt und unter „Mesh“ „IsoSurface“ ausgewählt (siehe Abb. 28).

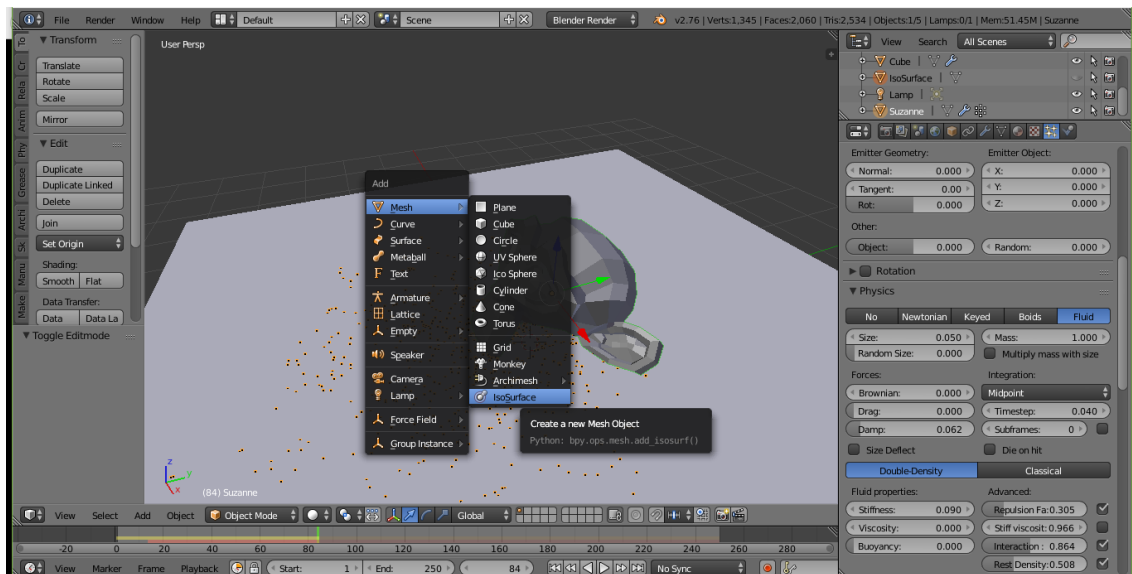


Abbildung 28: Um das Add-on zu benutzen, braucht es ein Partikelsystem und ein IsoSurface-Objekt.

Ist dieses Objekt erstellt, kann es im Outliner ausgewählt werden. Nun wird in dem Objekteigenschaften-Tab des Objektes unter „CubeSurfer Panel“ auf das Plus geklickt, um das Objekt zu benutzen. Daraufhin muss der Kopf (hier „Suzanne“) unter „Object“ und das Partikelsystem unter „Particle“ hinzugefügt werden. Anschließend wird in der Timeline, die sich am unteren Bildschirmrand befindet, auf den ersten Frame gewechselt, und auf „Set Local UV at this time“ geklickt. Nun sollte der Wert „Voxel size“ so lange weiter verringert werden, bis beim Abspielen der Animation eine wesentliche Veränderung des Erscheinungsbildes der Partikel zu sehen ist. Damit dies richtig funktioniert, sollte darauf geachtet werden, dass die Partikelgröße in den Partikelsystem-einstellungen größer als Null sein muss. Je nach Partikelsystem und gewünschtem Ergebnis können die Werte „Particles Size Multiplier“ und „Voxel Size“ eingestellt werden.

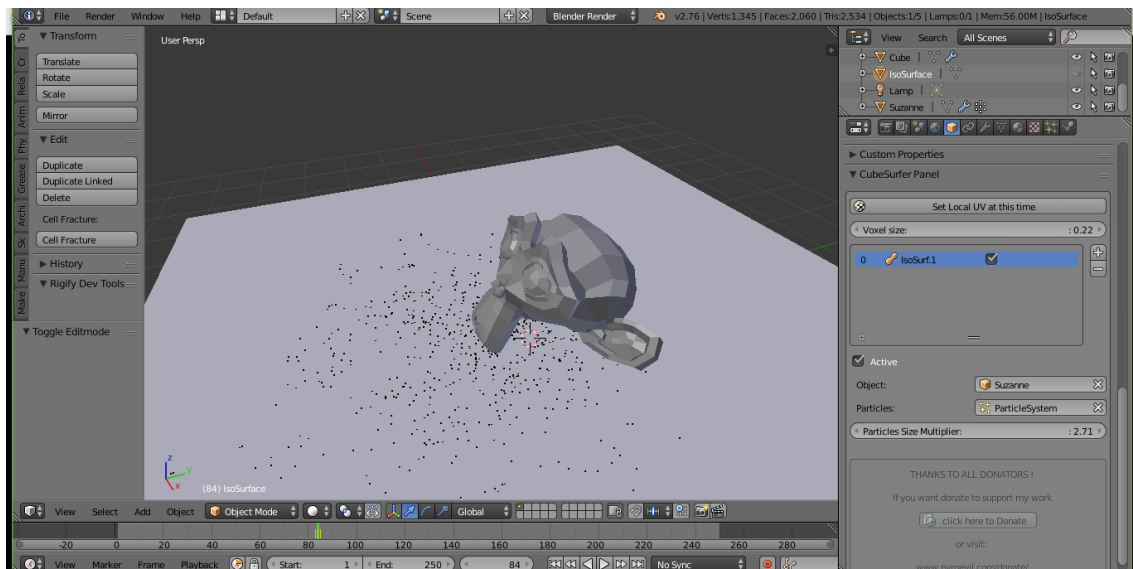


Abbildung 29: Das Objekt und dessen Partikelsystem wurden in dem CubeSurfer Panel eingefügt. Nun müssen die Werte so angepasst werden, dass aus den Partikeln eine Flüssigkeit entsteht.

Anschließend können die Werte des Partikelsystems weiterhin verändert werden, bis das gewünschte Ergebnis erzielt wurde. Leider kann nicht pauschal festgelegt werden, welche Werte für welche Flüssigkeit angegeben werden müssen, da sich die Flüssigkeit je nach Simulation anders verhalten soll. So ist langsam austretendes Blut anders zu erstellen, als solches, das bei einem Aufprall erzeugt wird, da hier einzelne Spritzer vom Kontaktpunkt aus abgegeben werden.

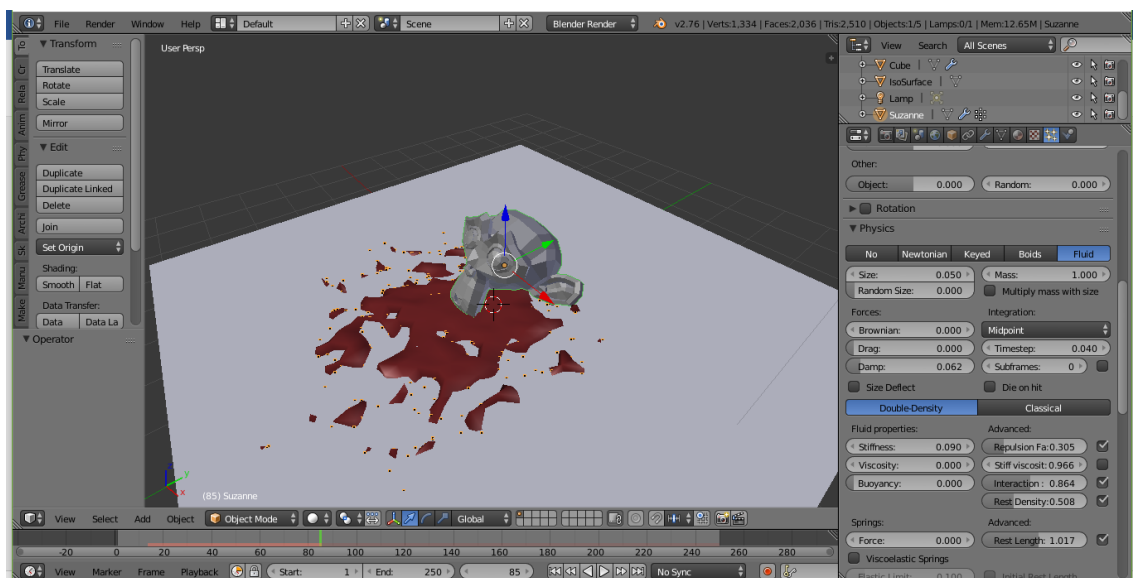


Abbildung 30: Auf der linken Seite sind die Einstellungen des Partikelsystems zu sehen. Diese wurden so eingestellt, dass sie in dieser Simulation zu dem gewünschten Ergebnis führen, müssen aber neu angepasst werden, wenn sich ein Umstand in der Szene ändert.

4.4 Simulieren der Tathergänge

Wird eine Tat begonnen, werden oft verschiedene Objekte zerstört, die Aufschluss über den Tathergang bringen können. Auch hier gibt es ein blenderinternes Tool um die Zerstörung verschiedener Gegenstände zu simulieren. Dies ist das „Cell-Fracture“ Add-on, das zwar schon in Blender integriert ist, aber erst in den User Preferences aktiviert werden muss. Ein anderes, externes Programm, das auf Blender aufgebaut ist, ist das „Fracture Modifier Build“ von Dennis Fassbänder. Dies kann mit einem Add-on, das ebenfalls von ihm erstellt wurde, erweitert werden. Das Add-on „Fracture Helpers“ und der Fracture-Modifier-Build können auf der Website von Dennis Fassbänder [URL-12] heruntergeladen werden.

In Blender ist die Physics Engine „Bullet Physics“ eingebaut, die nicht nur in Blender, sondern auch in anderen 3D-Programmen wie Maya oder 3Ds-Max verwendet wird. Sie ermöglicht die Interaktion verschiedener Objekte mit physikalischen Eigenschaften. So ist es möglich, verschiedenen Objekten unterschiedliche Massen und Oberflächeneigenschaften zu geben und diese anschließend unter Einfluss der Gravitation interagieren zu lassen. Diese Objekte werden „Rigid Bodys“ genannt, und sind all diejenigen Körper, die keine Flüssigkeiten oder Ähnliches sind. Beispielsweise ist ein menschlicher Körper, aber auch eine Wand oder eine Glasscheibe ein Rigid Body. Die Engine differenziert hier zwischen aktiven und passiven Rigid Bodys. Ein aktiver Rigid Body ist ein Körper, der mit anderen aktiv agieren soll, beispielsweise soll der Körper beim Aufprall auf den Boden abprallen. Auch soll die Gravitation, die bei Blender standardmäßig auf die Gravitationskraft der Erde eingestellt ist, auf diesen Körper wirken. Jedoch soll der Boden nicht „nach unten“ fallen, sondern stets starr bleiben, aber den aktiven Körpern die Möglichkeit bieten abzuprallen. Deswegen muss beispielsweise der Boden als passiv eingestellt werden. So kann er mit anderen Körpern interagieren, wird aber nicht aktiv in seiner Form oder Position in der Szene manipuliert.

Cell Fracture

Das Cell-Fracture-Add-on interagiert mit den physikalischen Einstellungsmöglichkeiten der Bullet-Physics-Engine. Zur Veranschaulichung werden ein Boden, eine Wand und ein kleiner Würfel benutzt, um das Zerstören einer Wand zu simulieren. Der kleine Würfel dient hierbei als Wurfgeschoss, und die Wand soll beim Aufprall des Würfels in kleine Stücke zerschellen. Die Wand soll eine einfache Steinwand darstellen.

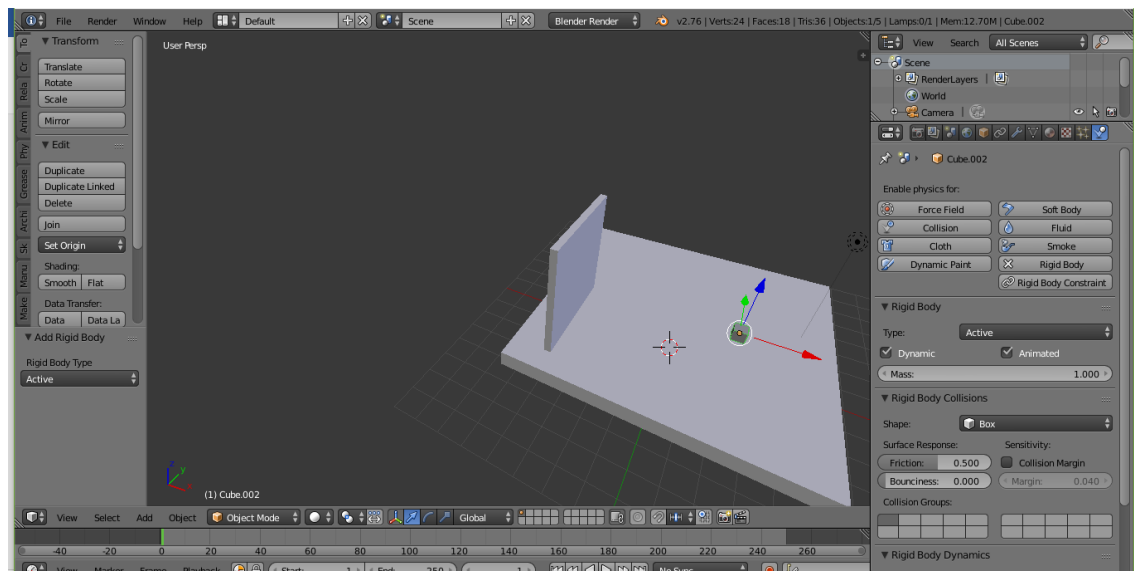


Abbildung 31: Bei dieser Beispielszene wurden ein Boden, eine Wand und ein kleiner Würfel erstellt. Nun sollen die richtigen Physik-Einstellungen an den Objekten vorgenommen werden. Hierzu wird der Physics-Tab in dem Property-Panel benutzt.

Im Property-Panel befindet sich der Physics-Tab, welcher meist der letzte Tab in dem Panel ist. Ist ein Objekt ausgewählt, können diesem verschiedene Eigenschaften zugeteilt werden (siehe Abb. 31). Der kleine Würfel soll ein aktiver Rigid Body sein und animiert werden. Hierzu wird auf die entsprechenden Buttons geklickt und im Drag-Down-Menü unter „Type“ „Active“ ausgewählt. Die Wand soll ähnliche Eigenschaften zugewiesen bekommen, jedoch soll diese nicht animiert werden, also wird kein Haken bei „Animated“ gesetzt. Da der Boden ein starres Objekt sein soll, mit dem später die einzelnen Bruchstücke der Wand interagieren sollen, wird dies auch ein Rigid Body, allerdings ein passiver.

Am unteren Bildschirmrand befindet sich der Player zum Abspielen der Animationen. Dieser wird später verwendet, um die Simulation abspielen zu lassen.

Damit die Wand in einzelne Stücke zerschellen kann, muss sie vorab in einzelne Teile zerschnitten werden. Auf der rechten Seite des Viewportes ist unter dem Tab „Tools“ nach Aktivierung des Add-ons nun ein Button namens „Cell Fracture“ zu finden. Um das Zerschellen einer Wand beim Kontakt mit dem Geschoss zu simulieren, reicht es aber nicht, wenn die Wand gleichmäßig zerspringt. Vielmehr sollten sich in der Nähe des Kontaktes beider Objekte kleinere und mehrere einzelne Bruchstücke befinden, die radial von der Kontaktstelle an Größe zunehmen und in der Anzahl abnehmen. Um dies zu realisieren kann der „Grease Pencil“ verwendet werden. Hierzu wird „N“ im Viewport gedrückt, daraufhin erscheint ein weiteres Panel mit neuen Einstellungen.

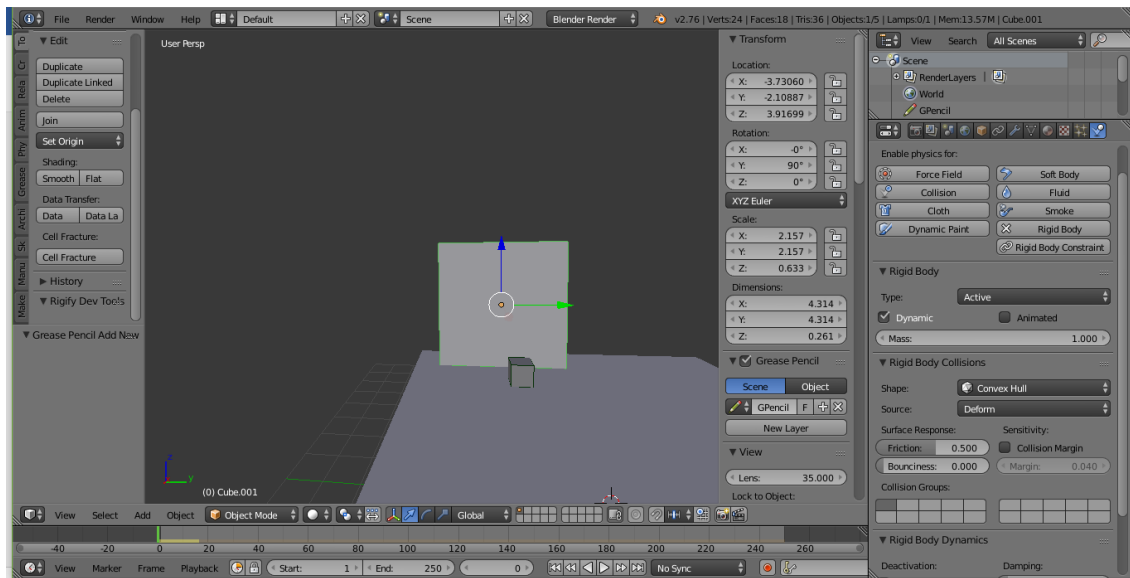


Abbildung 32: Wird im Viewport "N" gedrückt, erhält man weitere Einstellungsmöglichkeiten für das ausgewählte Objekt. Hier kann auch der Grease Pencil ausgewählt werden, und somit auf der Oberfläche des Objektes gezeichnet werden, wenn statt „Scene“ „Object“ ausgewählt wird.

Nun sollte „Object“ statt „Scene“ ausgewählt und durch Klick auf „New Layer“ eine neue Zeichenebene erstellt werden. Wird im Viewport „D“ gedrückt gehalten, ist es möglich etwas auf das ausgewählte Objekt zu zeichnen (siehe Abb. 33).

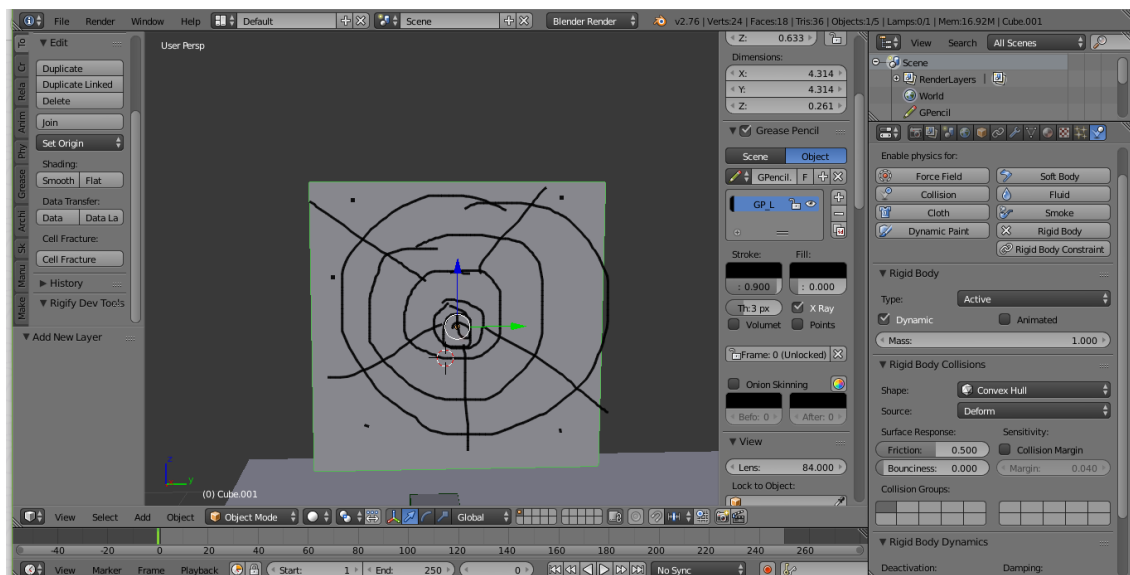


Abbildung 33: Hier wurde mittels des Grease Pencils ein Muster auf das Objekt gezeichnet. Dort, wo mehr Bruchstücke erwartet werden, wurden die Kreise enger und kleiner gezeichnet.

Da die Wand radial zerspringen soll, werden Kreise wie bei einem Spinnennetz angeordnet. Das Cell-Fracture Add-on wird die einzelnen Linien so übersetzen, dass sich zwischen ihnen die Bruchkanten befinden werden. Werden Punkte in größere blanke Teile der Zeichnung gesetzt, ist gewährleistet, dass diese nochmals zerteilt werden.

Nachdem das Bruchmuster grob definiert wurde, wird auf „Cell Fracture“ geklickt. Ein neues Pop-up-Fenster erscheint, dessen Einstellmöglichkeiten die Zerstückelung des Objektes beeinflusst.

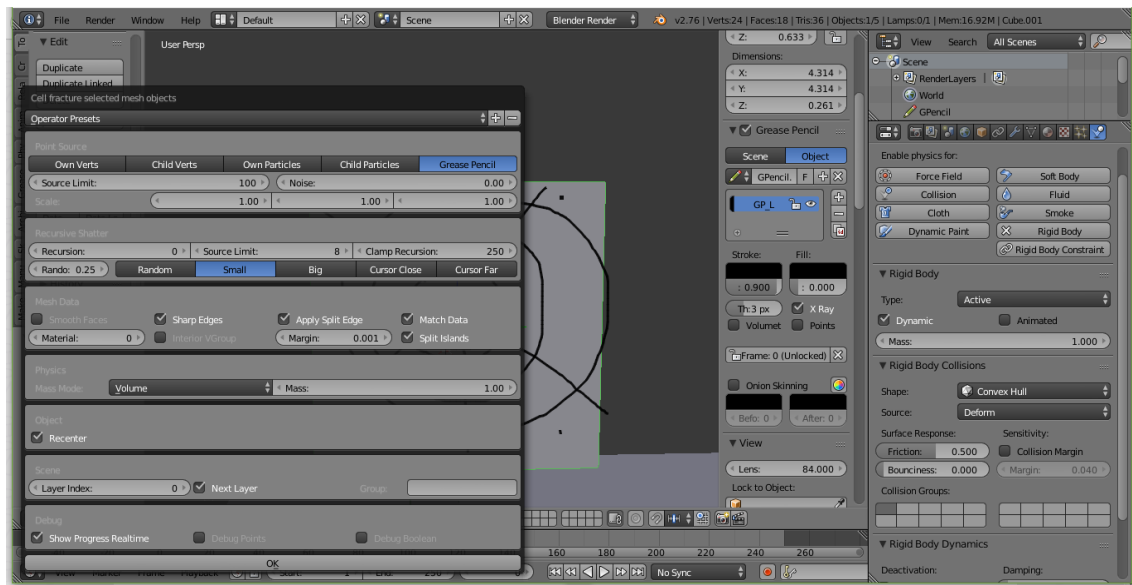


Abbildung 34: Das Pop-Up-Fenster gibt neue Einstellungsmöglichkeiten wieder. Verschiedene Berechnungsweisen der Fraktur sind möglich, und die Größe und Anzahl der einzelnen Stücke kann beeinflusst werden.

Um das Objekt unter Beeinflussung des angefertigten Musters zu zerschneiden, muss der Button „Grease Pencil“ ausgewählt werden. Unter „Source Limit“ kann eingestellt werden, wie viele einzelne Bruchstücke das Objekt zum Ende besitzen soll. Eine rekursive Zerteilung der Bruchstücke bedeutet, dass jedes einzelne Bruchstück nochmals zerteilt wird, je höher der Wert bei „Recursion“, desto öfter wird jedes Bruchstück rekursiv zerteilt. Dies kann jedoch zu Problemen und einer erhöhten Rechenleistung führen, und ist deshalb in diesem Beispiel nicht verwendet worden. Die anderen Werte können so gelassen werden, wie sie voreingestellt sind. Wird nun auf „OK“ gedrückt, dauert es einige Sekunden, bis auf dem zweiten Layer der Szene nun ein neues Objekt erscheint (siehe Abb. 35).

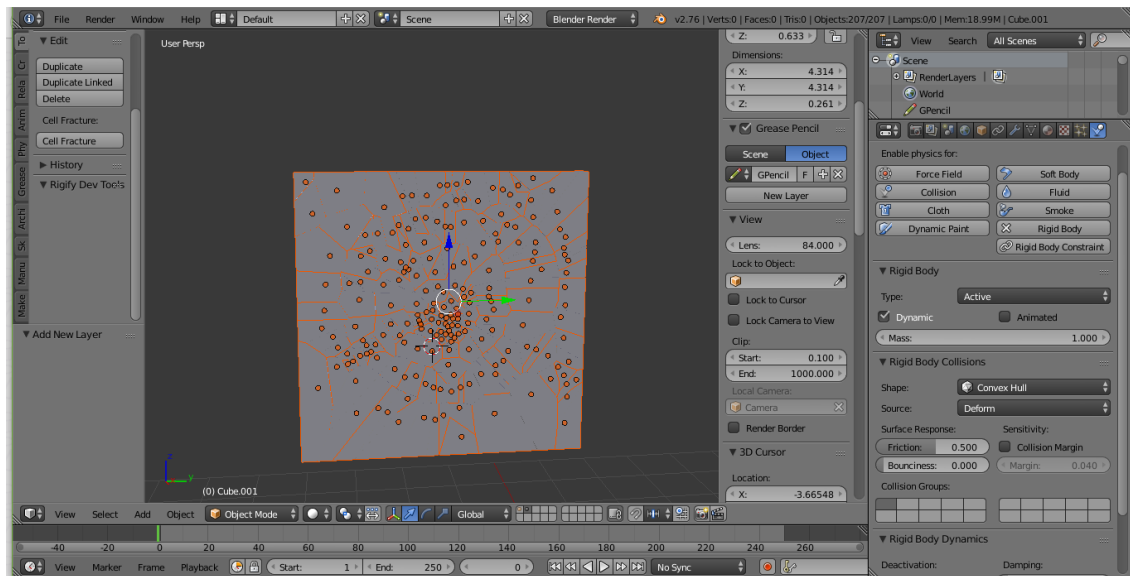


Abbildung 35: Das neue Objekt auf dem zweiten Layer der Szene (die Layer sind am unteren Bildschirmrand zugänglich) hat nun Bruchstücke als einzelne Objekte.

Damit es später zu keinen Problemen bei der Animation kommt, müssen doppelte Flächen entfernt werden. Dazu werden alle Bruchstücke ausgewählt (Shortcut „A“) und der Shortcut „Strg + J“ verwendet. Dies führt dazu, dass die einzelnen Teile wieder als ein Objekt zusammengeführt werden, aber die Kanten der Bruchstücke erhalten bleiben. Nun sollte die Tabtaste gedrückt werden, um in den Bearbeitungsmodus zu wechseln. Danach werden die doppelten Flächen mittels des Buttons „Remove Doubles“ unter „Tools“ im Tool-Panel entfernt. Um die einzelnen Stücke wieder voneinander zu trennen, muss „P“ gedrückt werden. Nun wird „Separate by Loose Parts“ ausgewählt. Danach wird per Tabtaste wieder in den Objektmodus gewechselt. Damit die Wand erst beim Aufprall zerschellt, wird ein Stück ausgewählt, und im Physic-Tab ein Haken unter „Rigid Body Dynamics“ bei „Enable Deactivation“ und bei „Start Deactivated“ gesetzt. Danach werden alle Stücke angewählt, und unter „Physics“ im Tool-Panel „Copy from Active“ angeklickt. Dadurch werden die Eigenschaften auf die anderen Stücke übertragen. Dies gewährleistet, dass das Objekt nur dann zerschellt, wenn ein anderes damit interagiert. Zuletzt muss die Masse der einzelnen Stücke korrigiert werden. Hierzu werden wieder alle Bruchstücke angewählt und unter „Physics“ im Tool-Panel auf „Calculate Mass“ gedrückt. Nun gibt Blender einem die Möglichkeit verschiedenen Voreinstellungen, wie Glass, Stein oder auch Gold zu verwenden (siehe Abb. 25).

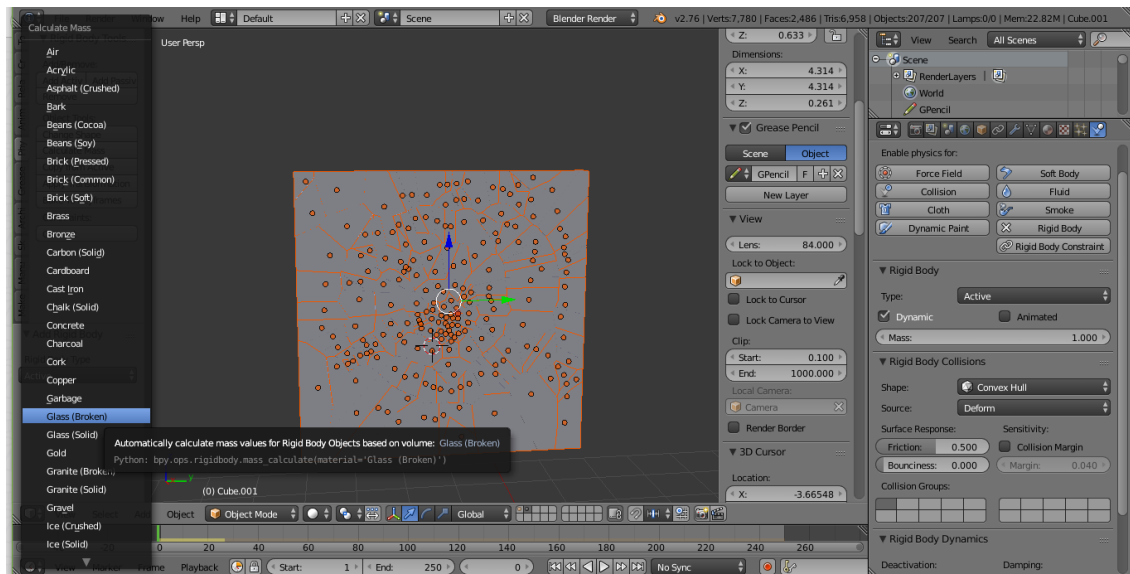


Abbildung 36: Blender bringt einigen Voreinstellungen zur Massenberechnung mit. Wird hier „Custom“ gewählt, kann die Dichte des Objektes selbst eingestellt werden.

Nun muss noch bewerkstelligt werden, dass das Objekt auch zerspringt, wenn der Würfel es berührt. Der Würfel muss sich dazu durch das Objekt hindurchbewegen. Dazu wird er animiert. Wird „i“ gedrückt, während der Würfel ausgewählt ist, kann ein Keyframe gesetzt werden. Ein Keyframe ist ein Schlüsselbild, das an einem bestimmten Punkt in der Zeitleiste einer Animation gesetzt wird. Dieses beinhaltet z.B. die Position des Objektes. Um den Würfel zu bewegen wird zuerst der erste Frame ausgewählt (siehe Abb. 36), dann der Mauszeiger über dem Objekt platziert. Nun wird „i“ gedrückt, und im Kontext-Menü „LocRotScale“ ausgewählt.

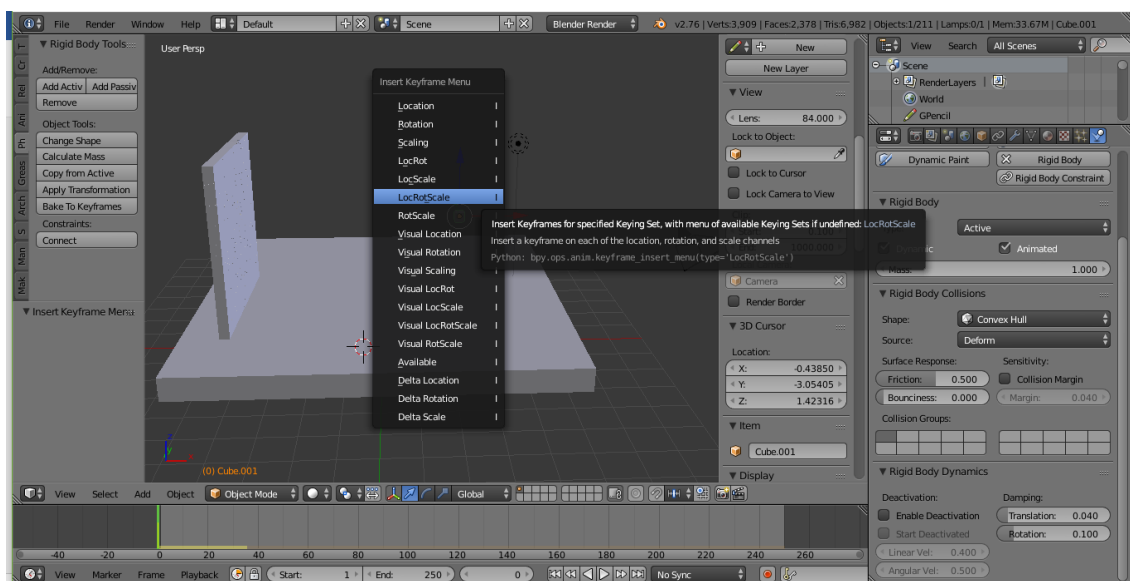


Abbildung 37: Setzen eines Keyframes

Dies gewährleistet, dass beim Frame Null die Position, Rotation und Skalierung des Würfels gespeichert wird. Nun muss der Frame gewählt werden, der den letzten Standpunkt des Würfels beinhaltet, zum Beispiel Frame 40. Bevor nun der nächste Keyframe gesetzt wird, muss der Würfel bewegt werden. Ist er an der richtigen Stelle, hier also hinter der Wand, kann der nächste Keyframe wie zuvor gesetzt werden. Wird nun die Animation abgespielt bewegt sich der Würfel von der rechten Bildschirmseite zur linken, und bewegt sich dabei durch die Wand. Diese zerbricht an der Aufprallstelle (siehe Abb. 38).

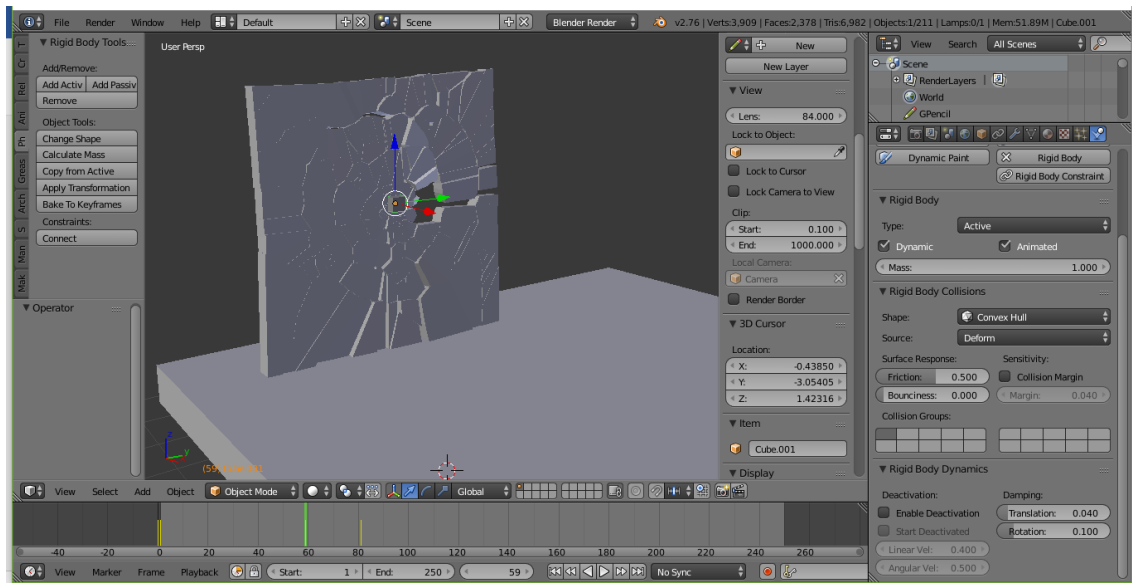


Abbildung 38: Die Wand zerbricht, nachdem der Würfel auf die Wand trifft.

Jedoch ist anzumerken, dass dies sehr aufwendig ist und leider, insbesondere bei komplexen Zerstörungen, noch zu Problemen führen kann. So passiert es manchmal, dass die einzelnen Teile der Wand in jede Richtung springen, was weniger realistisch scheint. Es ist also kompliziert für einen Laien mit Hilfe dieses Verfahrens ein realistisches Ergebnis zu erhalten.

Fracture Modifier Build

Nachdem Fracture-Modifier-Build heruntergeladen wurde, sollte das Add-on Fracture-Helpers ebenfalls installiert werden. Dies funktioniert wie die Installation der vorherigen Add-ons, nur muss dies in der modifizierten Blender Version von Fracture- Modifier-Build angewendet werden. Zur Veranschaulichung der Funktionsweise des Add-ons wurde hier die gleiche Szene aufgebaut.

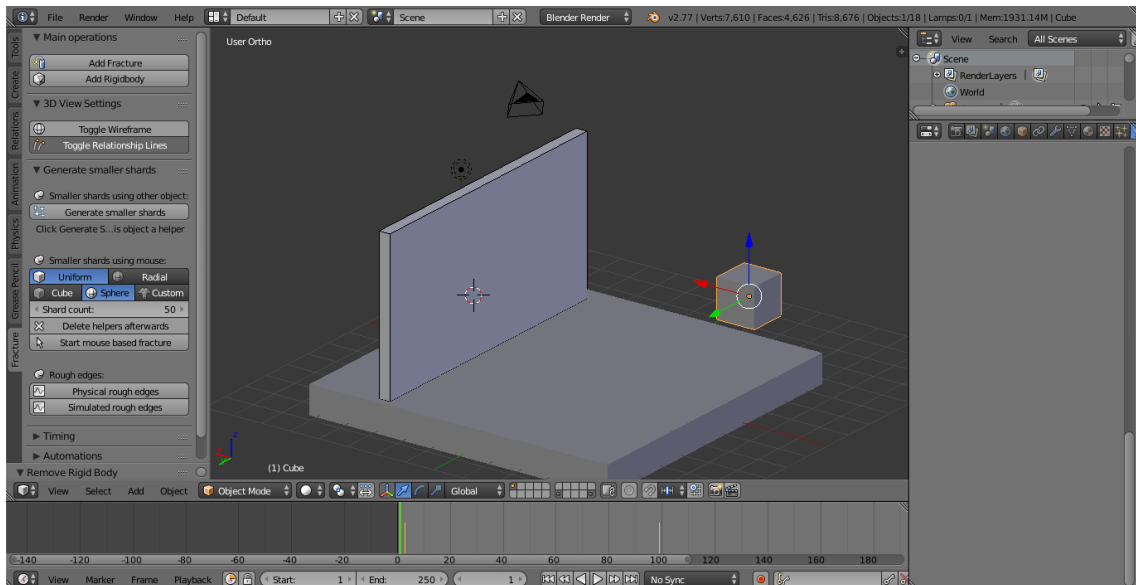


Abbildung 39: Die Szene besteht aus einer Wand, die zerstört werden soll, einem Boden und einem Würfel, der als Wurfgeschoss dient.

Im Tool-Panel befindet sich nun ein Tab namens „Fracture“, hier sind einige Grundfunktionen des Modifiers schnell zugänglich und einige Funktionen zusätzlich eingefügt. Ist der Würfel ausgewählt, kann hier z. B. sofort der Rigid-Body dem Körper hinzugefügt werden. Dies ist auch für Wand und Boden notwendig. Nachdem jedem Körper ein Rigid-Body zugewiesen wurde, kann auch definiert werden, ob dieser passiv oder aktiv ist (siehe Abb. 40). Dies wird wie beim Cell-Fracture-Add-on eingestellt. Sowohl bei der Wand als auch bei dem Würfel sollte zusätzlich ein Haken bei „Animated“ gesetzt werden. Eine Besonderheit des Modifiers ist, dass es einem die Möglichkeit gibt, einem Objekt eine „Trigger“- oder „Triggered“- Eigenschaft zu geben. Ein Trigger ist ein Objekt, das in diesem Kontext bei Interaktion mit einem anderen Objekt ein bestimmtes Verhalten auslöst. Das Objekt, mit dem der Trigger interagiert, ist das Triggered-Objekt, und reagiert auf das Eintreten des Triggers. Somit muss die Wand als „Triggered“ und der Würfel als „Trigger“ eingestellt werden. Um die Wand zu zerteilen, wird sie ausgewählt und „Add Fracture“ angeklickt.

Damit der Würfel nun die Wand durchkreuzt, sobald die Animation abgespielt wird, muss auch hier, wie zuvor erklärt mit Keyframes gearbeitet werden.

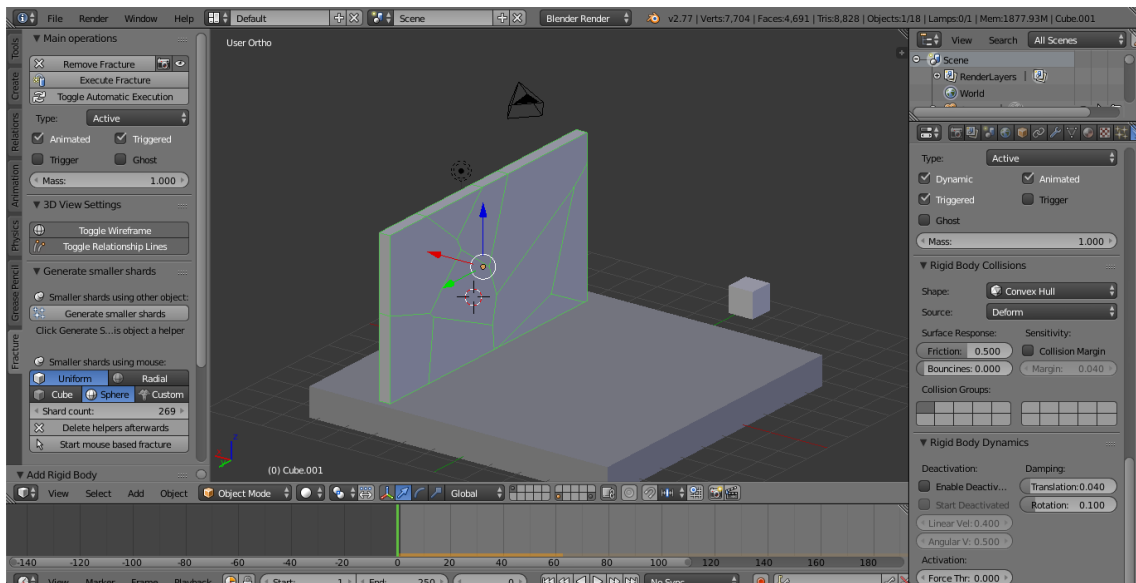


Abbildung 40: Die Wand wurde zerteilt, indem auf "Add Fracture" geklickt wurde. Nun kann im Properties-Panel unter „Physics“ die Fraktur dem gewünschten Ergebnis angepasst werden.

Nachdem das Objekt zerteilt wurde, kann im Properties-Panel unter „Shard Count“ die Anzahl der einzelnen Bruchstücke eingestellt werden. Auch das Aussehen der Bruchstücke kann beeinflusst werden, indem z.B. die Bruchstücke an der X-Achse entlang skaliert werden. Dies kann durch einen Klick auf „Splinter X“ und Eingabe der Länge des Bruchstückes realisiert werden (siehe Abb. 41). Nachdem die gewünschten Einstellungen getroffen wurden, muss die Wand nochmals zerteilt werden. Dies kann im Tool-Panel unter „Execute Fracture“ ausgeführt werden.

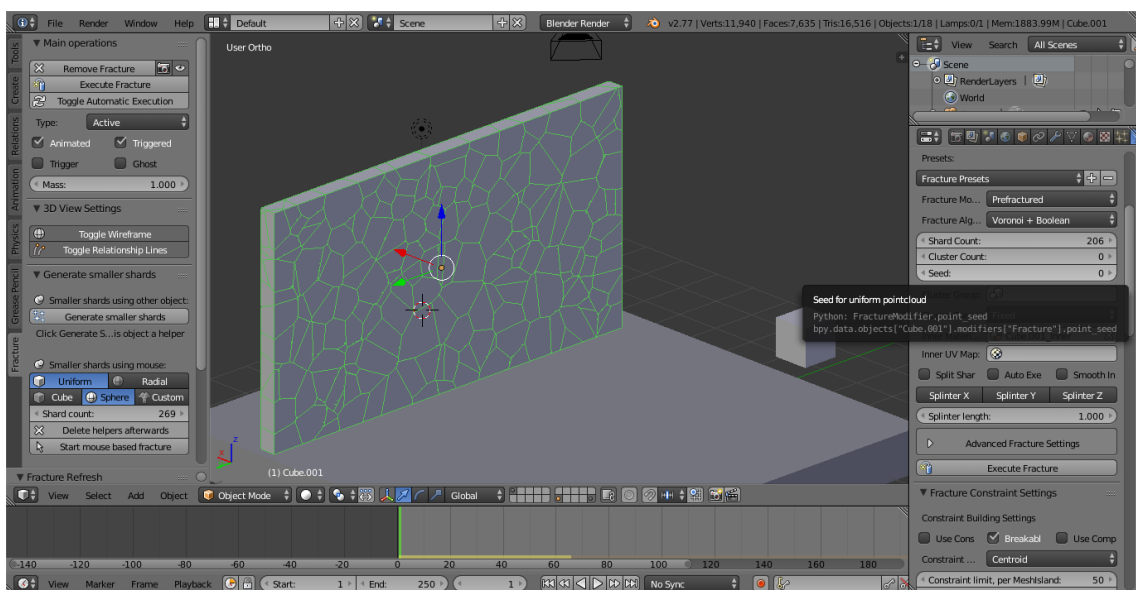


Abbildung 41: Die Anzahl der Bruchstücke wurde erhöht. Nun können Eigenschaften den Bruchstücken noch mehr zugeordnet werden.

Damit die Wand an der Kollisionsstelle zwischen Würfel und Wand in kleinere Stücke zerteilt wird, als am Rand der Wand, kann die Funktion „Generate smaller shards“ verwendet werden. Hierzu wird in der Zeitleiste an den Frame navigiert, an dem sich der Würfel und die Wand treffen (siehe Abb. 42). Nun wird zuerst der Würfel, dann die Wand ausgewählt (durch Shift + rechte Maustaste). Daraufhin wird der Button „Generate smaller shards“ im Tool-Panel angeklickt.

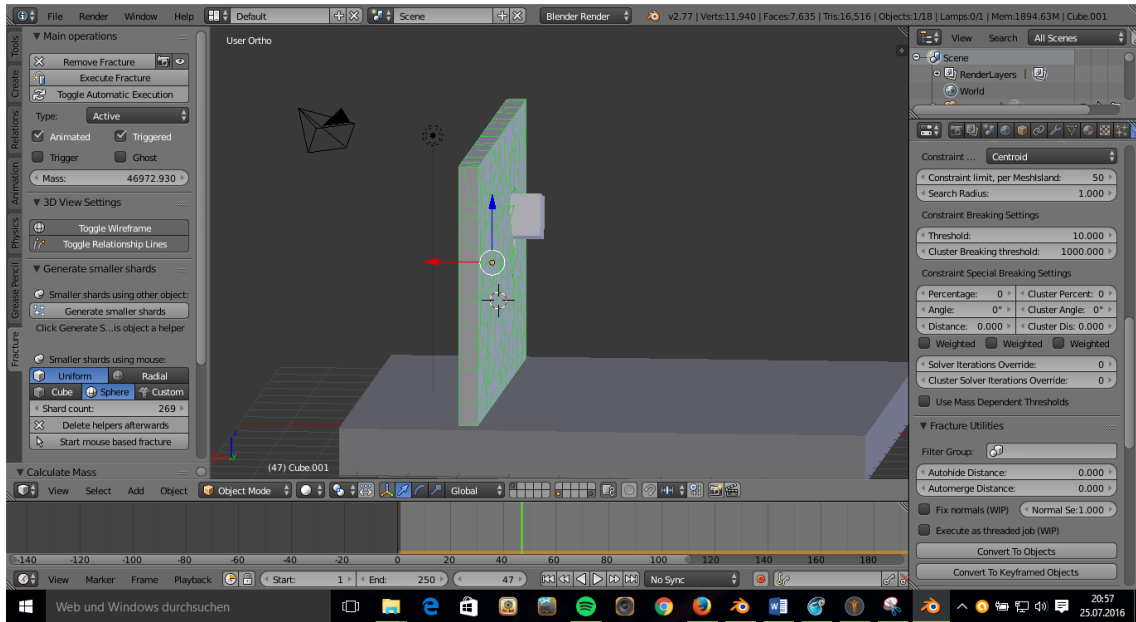


Abbildung 42: In diesem Beispiel treffen sich der Würfel und die Wand bei Frame 47. Mit der Funktion „Generates smaller shards“ kann nun ein Helfer-Partikelsystem erstellt werden, das die einzelnen Bruchstücke neu zerteilt.

Der Modifier erstellt nun auf dem letzten Layer ein Partikelsystem in der Form des Würfels, und generiert dadurch neue kleinere Bruchteile. Das Partikelsystem kann in seiner Größe und Zufälligkeit angepasst werden. Je größer das Partikelsystem, desto mehr kleinere Bruchteile werden an dieser Stelle generiert. Je größer der Wert unter „Random“, desto asymmetrischer werden die Bruchstücke angeordnet. Um dies einzustellen muss das Helfersystem ausgewählt werden. In Abb. 43 ist zu sehen, dass dieses nur durch eine grüne Box mit dem Titel „Cube_Helper“ dargestellt ist. Diese Box kann Skaliert oder auch Dupliziert (Shift + D) werden, um so noch genauerer Ergebnisse zu erzeugen. Jedoch hat die Skalierung dieser Box nur den Effekt, dass die Ausbreitung des Partikelssystems weiter gefasst ist, aber nicht die Anzahl der Partikel im System beeinflusst. Ist das Helfersystem ausgewählt, können im Tool-Panel die genannten Einstellungen vorgenommen werden. Um die Einstellungen auf die Wand zu übertragen, muss abermals „Execute Fracture“ ausgeführt werden.

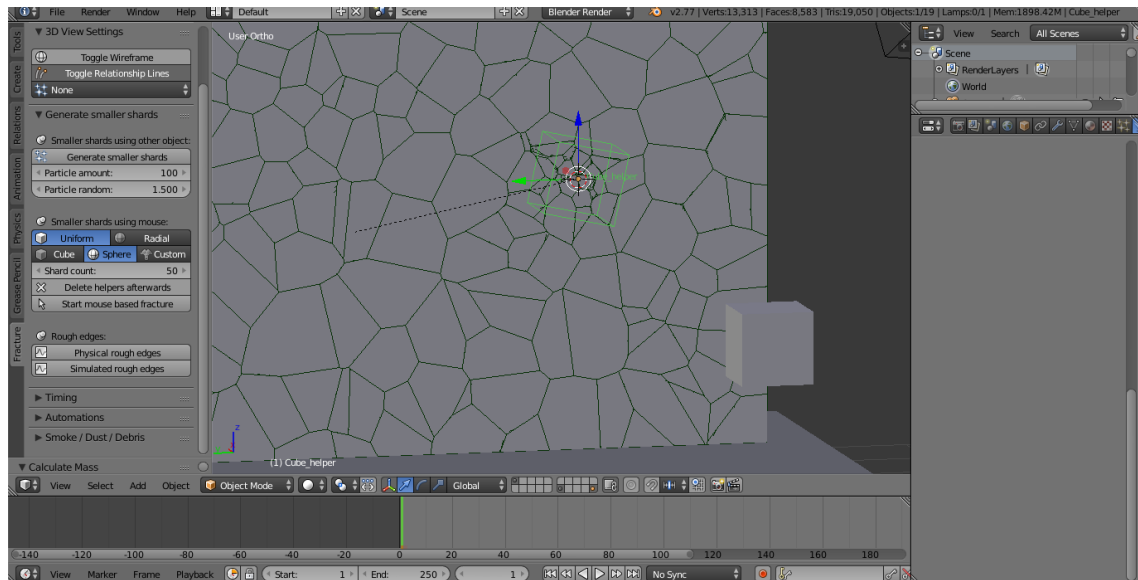


Abbildung 43: Durch das Helfersystem ist es möglich, die Fraktur besser an die gewünschten Ergebnisse anzupassen.

5 Auswahl geeigneter Add-ons

Nachdem nun mehrere Add-ons vorgestellt und in ihrer Funktionsweise gezeigt wurden, sollen sie nun miteinander verglichen werden. Hierbei wird vor allem auf die Nutzerfreundlichkeit eingegangen.

Das Fluid-Designer-Standalone und das Add-on Archimesh sind in ihrer Funktionsweise ähnlich. Beide können Räume erstellen, bei denen die Länge, Breite und Höhe angepasst werden können. Auch die Winkel zwischen den einzelnen Wänden können eingestellt werden. Jedoch können keine gekrümmten Wände im Fluid Designer erstellt werden. Des Weiteren ist die Erstellung der Wände bei beiden sehr unterschiedlich. Bei Archimesh kann zu Beginn der Raumerstellung sofort festgelegt werden, wie viele Wände der Raum am Ende besitzen soll. Diese können jederzeit angepasst werden, ohne dabei jede Wand einzeln anwählen zu müssen. Der Fluid-Designer hingegen hat keine Übersicht der Wände mit ihren Maßen im Viewport, deswegen müsste jede Wand einzeln korrigiert werden, sofern dies gewünscht ist.

Beide bringen die Möglichkeit, Fenster und Türen einzubinden. Hier ist der Fluid-Designer weitaus handlicher als Archimesh, da dieser diesen Prozess per Drag-And-Drop löst. Jedoch kann in Archimesh entschieden werden, in welche Richtung sich Fenster und Türen öffnen lassen, und dies könnte in einer Animation miteinbezogen werden.

Die Benutzung von Archimesh hat auch den Vorteil, dass es einfach in Blender integriert, und so mit anderen Add-ons gleichzeitig verwendet werden kann. Obwohl Archimesh in der Anwendung bei der Integration von Fenstern und Türen weitaus komplizierter ist, wird dennoch Archimesh für die Tatortrekonstruktion am Fallbeispiel gewählt, da das Add-on über mehr Einstellungsmöglichkeiten verfügt, im Allgemeinen übersichtlicher ist, sich leichter mit anderen Add-ons verwenden lässt und flexibler ist, wenn etwas verändert werden muss. Außerdem basiert der Fluid Designer auf Blender 1.74, wo hingegen die anderen hier vorgestellten Add-ons für höhere Blenderversionen erstellt wurden und nicht einwandfrei mit der Blenderversion 1.74 funktionieren.

Bei der Personenerstellung wurden zwei ähnliche Applikationen vorgestellt. Das MakeHuman-Standalone ist aber umfangreicher als das Add-on Manuel-Bastioni-Lab. So können in MakeHuman Personen im Alter von einem bis 75 Jahren erstellt werden, wohingegen im Blender Add-on nur Erwachsenen kreiert werden können. Des Weiteren können im Standalone Kleidung und Haare per Mausklick erstellt werden, was im Add-on nicht möglich ist. Zwar wurde vom makeHuman-Team für Blender eine Erweiterung (makeClothes) bereitgestellt, aber diese funktioniert nicht einwandfrei mit dem Manuel-Bastioni-Lab und ist somit nicht verwendbar.

Auch überzeugt MakeHuman bei der Erstellung der Personen. Das Standalone ist übersichtlich aufgebaut und mithilfe der Schieberegler kann schnell das gewünschte Ergebnis geliefert werden, wohingegen das Add-on zwar einen ähnlichen Umfang hat, jedoch schwieriger in der Handhabung ist. Es ist nicht auf den ersten Blick klar, welcher Wert auf welche Körperstelle Einfluss hat, was im Gegensatz dazu bei MakeHuman durch Illustrationen deutlich gemacht wurde. Das Manuel-Bastioni-Lab kann die Massenverteilung des Körpers und die Haut der Personen viel realistischer darstellen, jedoch ist ein solcher Detailgrad bei der Personenerstellung für die Tatortrekonstruktion nicht notwendig, und die Realistische Darstellung der Haut kann dazu führen, dass Blender langsamer wird.

Zwar ist es von Vorteil, wenn möglichst wenige Standalones benutzt würden, jedoch ist der Import und Export der Dateien von MakeHuman einfach. Insgesamt ist MakeHuman nutzerfreundlicher als das Manuel-Bastioni-Lab, weshalb im weiteren Verlauf MakeHuman zur Personenerstellung benutzt wird.

Bei der Visualisierung von Bewegungsabläufen kann meist das blenderinterne Partikelsystem verwendet werden. Dieses ist sehr umfangreich und schlüssig in der Bedienung. Jedoch stößt es an seine Grenzen, wenn es um die Simulation von Flüssigkeiten geht. Zwar ist es möglich, mithilfe von Partikelsystemen und der Physics-Einstellung „Fluid“ realitätsnahe Simulationen zu erstellen, jedoch muss hierzu das Partikelsystem „gebaked“ werden. Dies bedeutet, dass Blender das Verhalten der Partikel für die Länge der Simulation vorrechnen muss. Dies liefert meist gute Ergebnisse, ist aber zur schnellen Erzeugung unbrauchbar. Auch soll am Ende der Tatortrekonstruktion keine physikalisch akkurate Simulation von Blutspritzern stattfinden, sondern vielmehr eine aussagekräftige Visualisierung der Tathergänge. Hierzu kann das Cubesurfer-Add-on verwendet werden. Dies kommt ohne „baken“ aus und verwendet die Physics der Partikelsysteme.

Zwischen dem blenderinternen Cell-Fracture-Add-on und dem Fracture-Modifier-Build gibt es einige Unterschiede. Das Fracture-Modifier-Build ist übersichtlicher aufgebaut und effektiver in der Handhabung als das Add-on. So sind die wichtigsten Einstellung durch die Erweiterung mit dem Fracture-Helpers-Add-on, in dem Tool-Panel aufgelistet. Auch ist es möglich die Objekte als „Trigger“ oder „Triggered“ einzustellen, wodurch der Aufbau der Simulation vereinfacht wird. Des Weiteren ist der Fracture-Modifier-Build schneller und stabiler in der Verarbeitung der Animation. Auch kommt es zu keinen Problemen bei der Fraktur durch „Doubles“ wie in Cell-Fracture.

Durch die einfache Handhabung und der besseren Stabilität wird in dem Fallbeispiel auf den Fracture-Modifier-Build zur Visualisierung von Zerstörungen zurückgegriffen.

6 Anwendung geeigneter Add-ons auf ein Fallbeispiel

Das Fallbeispiel ist basierend auf dem Doppelmord der in 3. vorgestellt wurde. Die Anordnung der einzelnen Räume ist hier aber fiktiv, da kein Grundriss des Tatortes zur Verfügung steht. Demnach wurde ein fiktiver Grundriss (siehe Abb. 44) erstellt, der sich an die Erklärungen aus „Tatorte 2“ anlehnt.

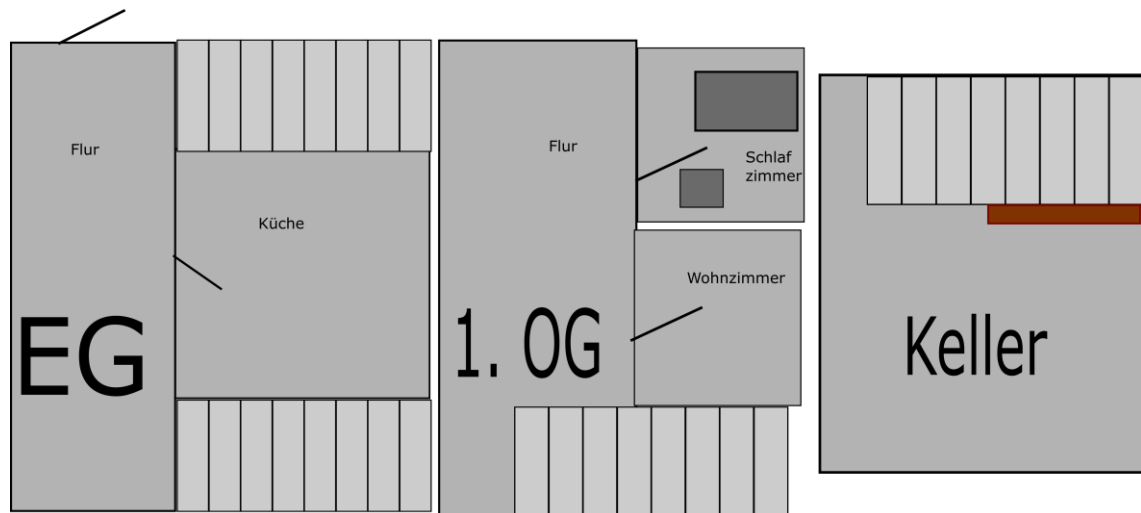


Abbildung 44: Maßstabslose Darstellung des Tatortes.

Dieser Grundriss soll für die Tatorterstellung mittels Archimesh dienen. Frank Nicolaus beschrieb Elsa Tomschke als 54 Jahre alte Frau, die 159 cm groß und 85 kg schwer war. Paul Tomschke soll 171 cm groß, 93 kg schwer und 72 Jahre alt gewesen sein. Da beide aus Sachsen kamen, wird angenommen, dass beide europäischer Herkunft waren. Der Täter, Klaus Schuhricht, wird als 23 Jahre alt beschrieben. [Ni12]

Anhand des Grundrisses wird in Archimesh das Gebäude erstellt. Dabei können zur Übersicht die einzelnen Stockwerke auf verschiedene Ebenen verschoben werden (Shortcut „M“). Auch können den einzelnen Wänden und Böden verschiedene Farben gegeben werden, indem im Properties-Panel ein neues Material hinzugefügt wird und dessen Farbe geändert wird. Um einzelne wichtige Objekte, wie Betten oder Tische einzufügen, wurde hier ein Skript (s. Anhang) verwendet, das den Import von einzelnen Objekten erleichtert. Hierzu wurde eine Objektbibliothek aus lizenzfreien 3D-Objekten erstellt [URL-13].

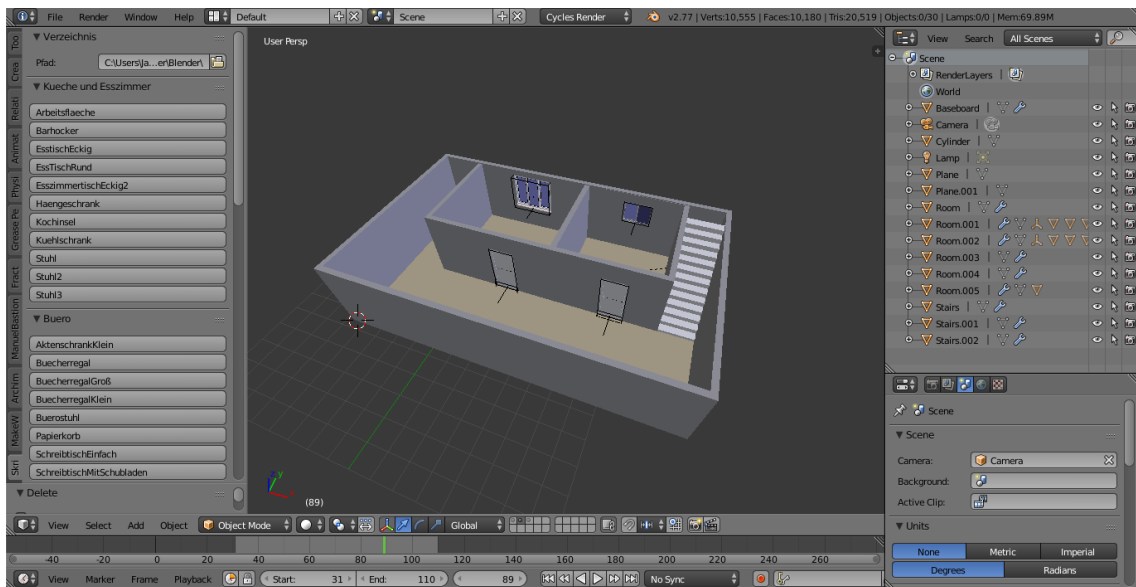


Abbildung 45: Das 1. Obergeschoss wurde mit Archimesh erstellt. Türen und Fenster wurden ebenfalls mittels Archimesh eingefügt. Nun werden noch einzelne wichtige Objekte wie Tische und Stühle durch ein Skript in die Szene eingefügt.

Anschließend werden allen Objekten die passenden Physikeigenschaften durch den Fracture-Modifier-Build und das Fracture-Helpers Add-on zugewiesen.

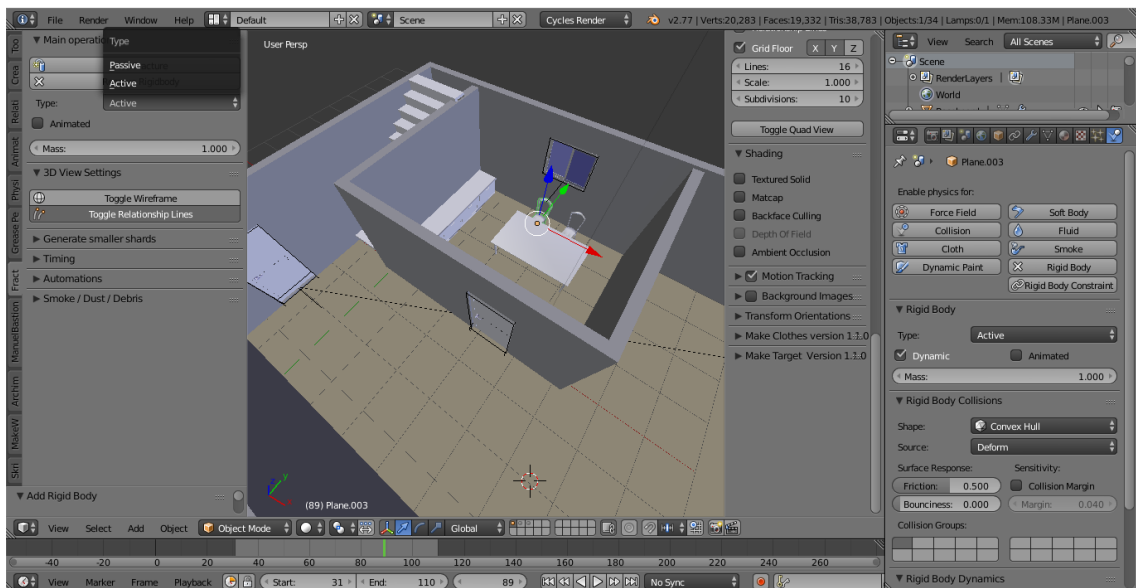


Abbildung 46: Allen Objekten wird ein Rigid-Body zugewiesen. Dadurch ist gewährleistet, dass im späteren Verlauf beispielsweise Blut, das durch das CubeSurfer-Add-on realisiert wird, haften bleibt.

Die involvierten Personen werden alle mit dem makeHuman-Standalone erstellt. Ihnen wird ein Motion-Capture-Skelett hinzugefügt, um anschließend als Collada-Objekte exportiert und in Blender importiert zu werden.



Abbildung 47: Von links nach rechts: Paul Thomschke, Elsa Thomschke und Klaus Schuricht. Die Klamotten und Haare wurden in makeHuman hinzugefügt.

Nun werden die Personen in der Szene platziert. Hierzu wird Frau Thomschke in der Küche, Herr Thomschke im Schlafzimmer im 1. Obergeschoss und Klaus Schuricht im Erdgeschoss aufgestellt.

Damit sich beispielsweise Klaus Schuricht zur Küche bewegt, muss ein Pfad, an dem er entlanglaufen soll, erstellt werden. Hierzu wird ein „Curve“-Objekt des Typs „Path“ benutzt (siehe Abb. 48).

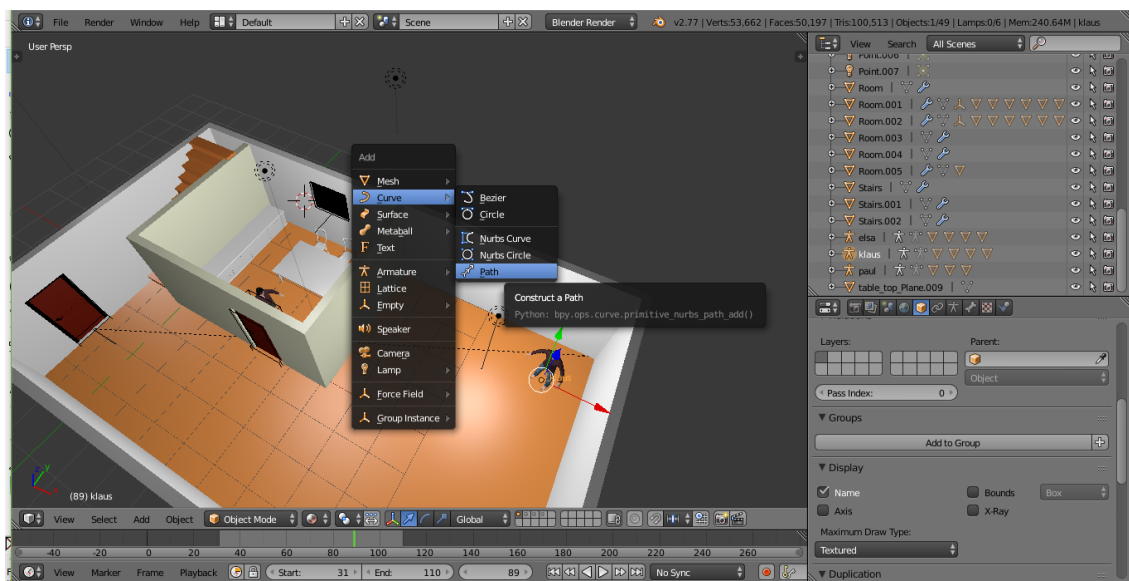


Abbildung 48: Das Path-Objekt dient als Pfad, an dem sich Klaus Schuricht später entlangbewegen soll. Um das Objekt zu erzeugen, wird der Shortcut „Shift + A“ verwendet und unter „Curve“ „Path“ ausgewählt.

Der Pfad kann nun im Bearbeitungsmodus angepasst werden. Danach wird dem Objekt, dass sich entlang des Pfades bewegen soll, im Properties-Panel ein „Constraint“ hinzugefügt. Dieses heißt „Follow Path“. Nachdem dieser Constraint hinzugefügt wurde wird unter „Target“ der Pfad ausgewählt und abschließend auf „Animate Path“ geklickt. Wird nun die Animation abgespielt, bewegt sich das Objekt entlang des Pfades.

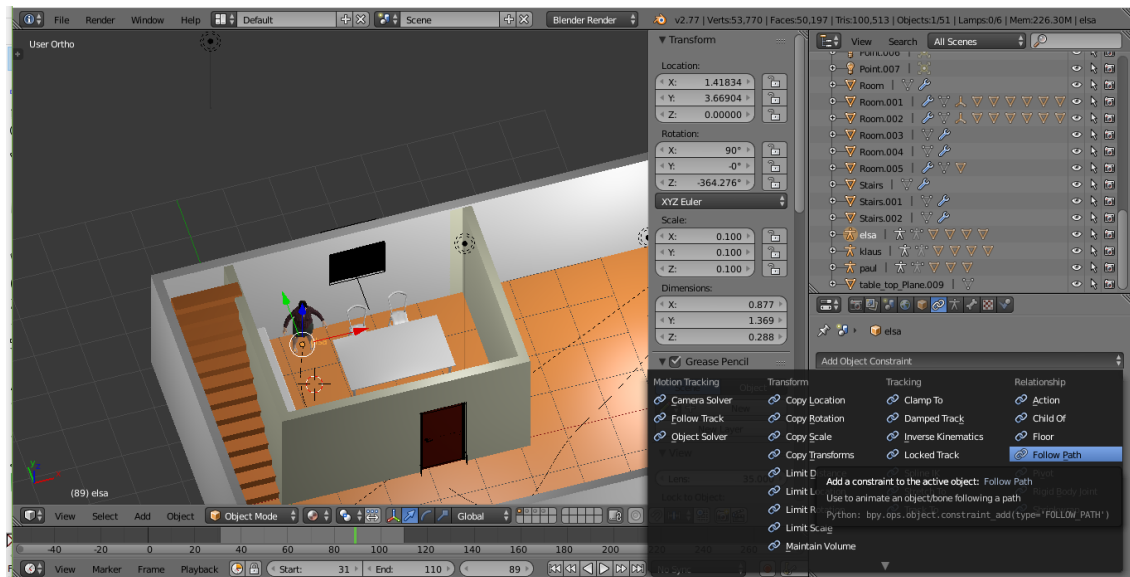


Abbildung 49: Damit sich die Person am Pfad entlang bewegt, muss der Constraint „Follow Path“ hinzugefügt werden. So kann auf einfache Weise die Person animiert werden.

Durch das Add-on „MakeWalk“ kann nun noch eine Laufanimation hinzugefügt werden. Hierzu wird das Add-on wie in 4.3 beschrieben, verwendet. Anschließend muss in den Dope-Sheet-Editor gewechselt werden (siehe Abb. 50). Hier kann nun in den Action-Editor gewechselt werden.

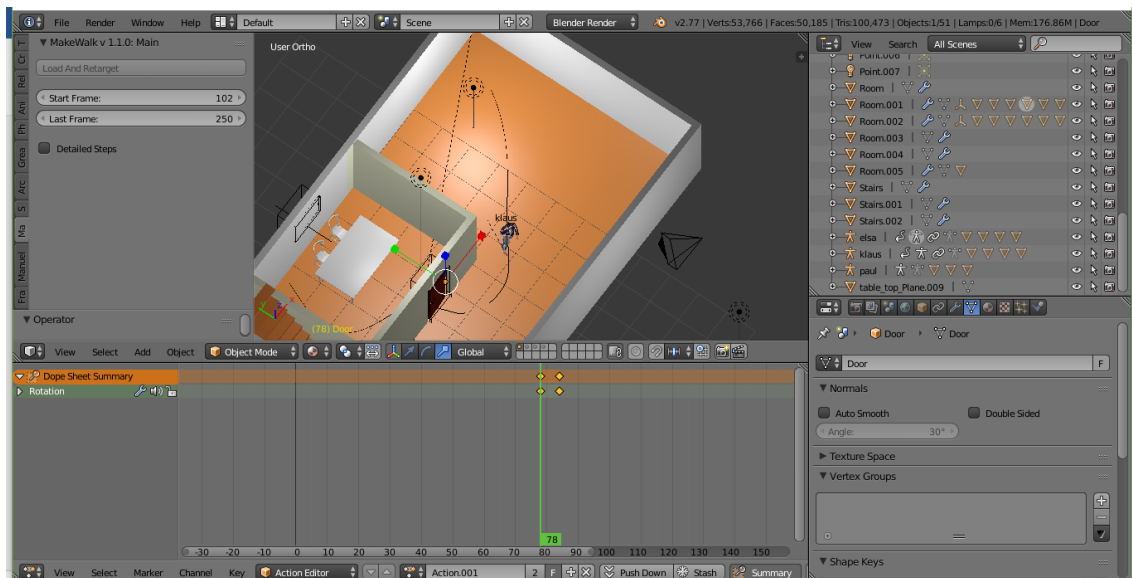


Abbildung 50: Indem eine Animation als Aktion festgelegt wird, hier die Animation der Tür, kann sie im NLA-Editor weiterverwendet werden. Dies ist nützlich, um Animationen zu wiederholen und wiederzuverwenden.

Um Animationen, die nicht durch makeWalk erstellt wurden, zu verwenden, müssen diese als „Action“ festgelegt werden. Hierzu wird am unteren Bildschirmrand neben dem Namen der Aktion auf den Knopf „F“ geklickt. Wird nun in den NLA-Editor gewechselt (siehe Abb. 51) kann diese Aktion hinzugefügt werden. Soll die Animation wiederholt werden, kann diese dupliziert oder verlängert werden. Jede Animation, die durch makeWalk erstellt wurde, wird automatisch als Aktion festgelegt. Im NLA-Editor können diese nun beliebig aneinandergereiht werden.

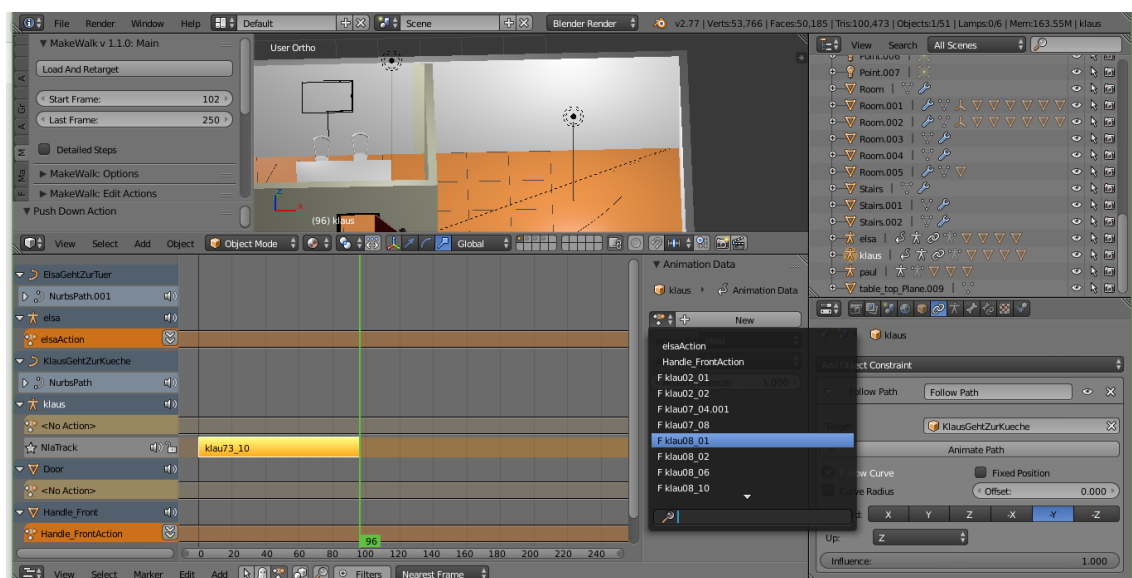


Abbildung 51: Um den NLA-Editor optimal zu nutzen, sollte der Shortcut „N“ gedrückt werden. Nun können an der rechten Seite des Editors die Aktionen ausgewählt und bearbeitet werden.

Da in diesem Fallbeispiel keine Zerstörungen stattfinden, wird hier nicht der Fracture-Modifier-Build benutzt. Jedoch kommt es zu mehreren Blutspritzern, die durch das CubeSurfer- Add-on realisiert werden können. Hierzu wird wie in 4.3. vorgegangen.

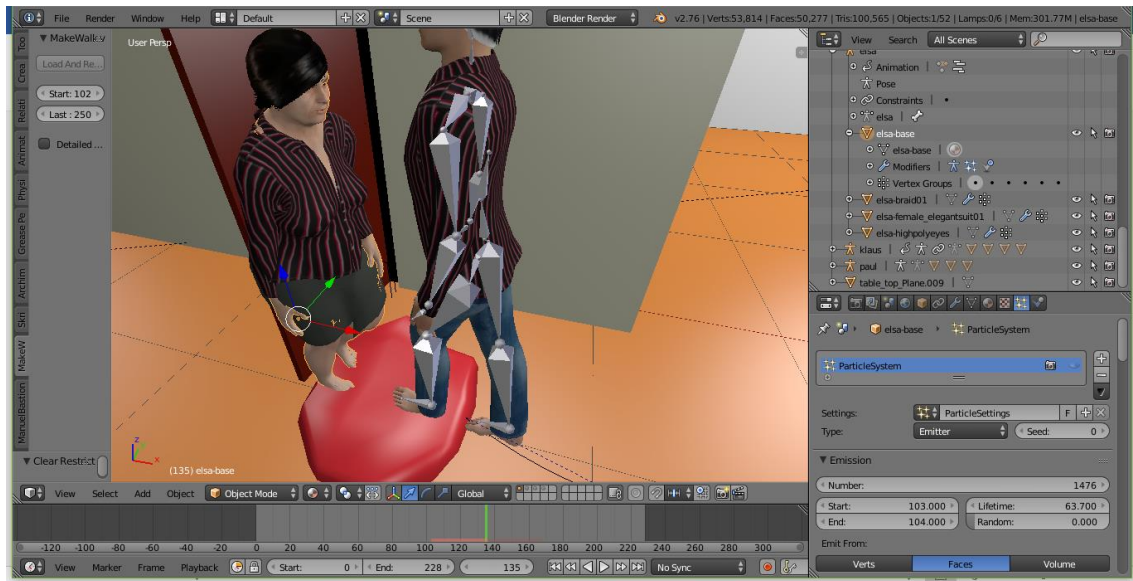


Abbildung 52: Mittels dem CubeSurfer-Add-on werden Blutspritzer und -lachen erstellt. Diese können dann je nach gewünschtem Ergebnis weiter angepasst werden.

Das Erstellen einer solchen Szene kann im Zeitaufwand variieren. Je nachdem, welche Bewegungsabläufe relevant für die Tatortrekonstruktion sind, addiert sich der Aufwand. Das hier vorgestellte Fallbeispiel konnte dank der Add-ons innerhalb weniger Stunden erstellt werden.

7 Diskussion

Durch die Analyse und Prüfung einzelner Add-ons für die Opens-Source-Software Blender sollte festgestellt werden, ob ein nützliches Softwarepaket erstellt werden kann, welches die computergestützte Tatortrekonstruktion unterstützt.

Hierzu wurden verschiedene Add-ons betrachtet und geprüft. Das Add-on Archimesh wurde zur Anfertigung des Grundrisses des Tatortes benutzt und erwies sich als hilfreich. Durch dessen Verwendung konnte nicht nur Zeit gespart werden, sondern auch die händische Modellierung vernachlässigt werden. Auch das Standalone makeHuman konnte mit einer übersichtlichen Benutzeroberfläche die Personenerstellung erleichtern. Das CubeSurfer-Add-on konnte zur Visualisierung von Blutlachen und -spritzern verwendet werden. Dies stützte sich auf die Bullet-Physics-Engine, die in Blender integriert ist. Als letztes hilfreiches Add-on wurde der Fracture-Modifier-Build vorgestellt, durch welches es durch wenige Klicks möglich ist, komplexe Zerstörungen zu simulieren.

Jedoch sind Blender und die vorgestellten Add-ons nicht explizit für den forensischen Gebrauch entwickelt worden. So sind beispielsweise die Bruchanimationen nur eine Simulation der Realität und spiegeln nicht das reale Zerschlagen eines Objektes wie Glas oder Stein wieder. Dies ist so, weil die 3D-Software insbesondere für die Animations- und Spielebranche verwendet wird und auch entwickelt wurde [URL-14]. Das bedeutet aber nicht, dass es nicht möglich ist, eine reale Simulation mit Open-Source-Tools zu bewerkstelligen. Schließlich wird die Software stets weiterentwickelt.

Ziel war es, eine für jeden Benutzer zugängliche Lösung zu finden, die eine repräsentative Tatortrekonstruktion zulässt. Dies ist nicht nur in Blender möglich. Beispielsweise beschäftigten sich Elissa St. Clair, Andy Manoney und Albert Schade III mit der Tatortrekonstruktion in SketchUp. [CL12] Auch hier wurde ähnlich vorgegangen. So wurde ein Tatort anhand einzelner Skizzen und Fotografien nachgebaut und anschließend in SketchUp modelliert, mit Objekten ausgestattet und ein Opfer in der Szene platziert. Auch hier war Ziel der Arbeit, eine kostenfreie Alternative zu teuren speziell für die Tatortrekonstruktion entwickelten Programmen zu finden. [CL12] Jedoch wurde hier nur die Modellierung des Tatortes behandelt. Es entstand am Ende ein Tatort, der virtuell begehbar war. Zwar wird in der Arbeit darauf hingewiesen, dass die Software durch Plug-Ins erweitert werden kann, doch muss hierzu die kostenpflichtige Pro-Version erworben werden. Dies unterstützt die Verwendung von Blender, da Blender im Gegensatz zu SketchUp, das zur Visualisierung von Architektur und Innenarchitektur verwendet wird, über die Modellierung einzelner Objekte hinausgeht. So besitzt es eine Physics-Engine und die Möglichkeit, alle Objekte zu animieren.

Nick V. Flor beschäftigte sich ebenfalls mit der digitalen Tatortrekonstruktion [Fl11]. In seiner Arbeit konzentrierte er sich auf die Modellierung in 3DsMax, welches kostenfrei zum akademischen Gebrauch verwendet werden kann, und bildete einen Tatort nur mit primitiven Objekten und händischer Modellierung nach. Er schrieb in seiner Zusammenfassung, dass zur Vervollständigung der Rekonstruktion vier weitere Dinge fehlen würden. So würden Charakter, die anhand von Fotografien realistisch nachgemodelt werden würden, realistische Texturen der Objekte, Animationen der Objekte, durch das Hinzufügen einzelner Skelette, und Skripte, die Animationen und Bewegungen auslösen noch fehlen. [Fl11] Hier zeigt sich, wie hilfreich die Add-ons für Blender sind. So können dank makeHuman die einzelnen Personen realistisch anhand von Fotoreferenzen nachgemodelt, mit Texturen versehen und auch sofort mit einem Skelett ausgestattet in die Szene importiert werden. Des Weiteren können auch Daten eines Laserscanners in Blender eingefügt werden. [Ko11] Auch Animationen sind auf einfache Weise in Blender möglich. Fracture-Modifier-Build kommt auch mit der Funktion mit, Objekte als „Trigger“ oder „Triggered“ festzulegen, welche ermöglicht, Animationen auszulösen. Wird dies noch mit dem Add-on makeWalk verbunden, kann nahezu jeder Bewegungsablauf realisiert werden ohne selbst zu animieren.

Jedoch kommt es auch zu Komplikationen, wenn Blender verwendet wird. Beispielsweise kann es zu Versionsproblemen kommen, wenn verschiedene Add-ons benutzt werden sollen. Da Blender laufend aktualisiert wird, und auch die Programmiersprache Python sich dauernd erweitert, können manche Add-ons nicht für die aktuellste Version verwendet werden. Dennoch ist dies im Vergleich zu den Vorteilen, die die Benutzung von Blender mit sich bringt, nur ein geringfügiges Problem. Schließlich kann dennoch ein repräsentatives Ergebnis ohne Kosten erzielt werden.

8 Zusammenfassung und Ausblick

Wie schon in Kapitel 7 besprochen, werden Blender und dessen Add-ons laufend entwickelt. So ist darauf zu hoffen, dass in späteren Blenderversionen bessere Ergebnisse und für den Laien benutzerfreundlichere Bedienungsmöglichkeiten realisierbar sind. Auch wird die Physics Engine weiter verbessert und mit der Weiterentwicklung der Technik sind vielleicht schon in einigen Jahren die Ressourcen geschaffen, realistische Simulationen zu erstellen. Diese erfordern eine hohe Rechenleistung, die meist durch Computer für den alltäglichen Gebrauch noch nicht abdeckt werden können.

Dennoch kann schon jetzt durch die hier vorgestellte Software die Tatortrekonstruktion erleichtert werden. Ein großer Vorteil ist hier, dass die Add-ons die händische Modellierung übernehmen und so selbst Laien einen Tatort und die involvierten Personen erstellen können. Da die Entwicklung der Add-ons meist von Privatpersonen aus eigenem Interesse geleitet werden, können auch hier in den nächsten Jahren Add-ons hinzukommen, die für die Tatortrekonstruktion zu gebrauchen sind. Beispielsweise wäre es hilfreich, ein Add-on wie CubeSurfer mit einzubeziehen, dessen Benutzung etwas einfacher und automatisierter ist, da sich der Benutzer mit den Partikelsystemen intensiv beschäftigen muss um ein gutes Ergebnis zu erzielen.

Auch schrieb Ton Roosendal, Chairman der Blender Foundation, dass Autodesk, Netflix und diverse Animationsstudios sich vermehrt mit Blender beschäftigen. [URL-15] Sollte eine Zusammenarbeit mit Autodesk stattfinden, ist es möglich, dass Blender intensiver von großen Animationsstudios verwendet wird, und somit sich das Budget, das nur aus Spenden besteht, sich erhöht, und dadurch die Software weiter verbessert werden kann. Dennoch ist eine Vorhersage zu der Entwicklung schwierig, da Blender ein Open-Source-Projekt, und somit von der Mitwirkung der Benutzer abhängig ist.

Durch Arbeiten wie die von Nick V. Flor oder Elissa St. Clair u. A. (siehe 7.) wird aber deutlich, dass Bedarf an kostenfreier Software zur Tatorterstellung besteht. So war die hier erläuterte Vorgehensweise ein weiterer gelungener Versuch, ein kostenfreies, benutzerfreundliches Softwarepaket zur digitalen Tatortrekonstruktion zu erstellen.

Literaturverzeichnis

- [Bu10] Buck, Ursula: Laserscanning in der Kriminalistik, in: zvf, 135, 3 (2010), S.190 -198
- [Ca15] Caudron, Romain / Nicq, Pierre-Armand: Blender 3D By Example (E-Book). 2015.
- [Cl12] St. Clair, Elissa u. a.: An Introduction to Building 3D Crime Scene Models Using Sketch Up, in: J Assoc Crime Scene Reconstruction, 18, 4 (2012), S. 29-47.
- [Fl11] Flor, Nick V.: Virtual Crime Scene Reconstruction. The Basics of 3D Modelling, in: Journal of Digital Forensics, Security and Law, 6, 4 (2011), S. 67-74.
- [Ko11] Komar, Debra A. u. a.: The Use of a 3-D Laser Scanner to Document Ephemeral Evidence at Crime Scenes and Postmortem Examinations, in: Journal of Forensic Sciences, 2011.
- [Kw07] Kwietnewski, Marcin u. a.: A multimedia database system for 3D crime scene representation and analysis, PDF, 2007.
- [Ni12] Nicolaus, Frank: Das Drama Im „Hirsch“ Zu Radeburg, in: Schade, Thomas / Schlinzig, Karsten (Hrsg.): Tatorte 2. Sächsische Kriminalfälle aus vier Jahrzehnten, Dresden 2012, S. 23 – 44.
- [St08] Steinert, Ulf: Grundlagen der Kriminaltechnik, Brandenburg, Fachhochschule, Kriminalistik/Kriminaltechnik, Skriptum, 2008
- [St12] Stiller, Heiner: Blender 2.5. 3D-Modellierung und -Animation (E-Book), 2012
- [Th05] Thali, Michael J. u. a.: VIRTOPSY-Scientific Documentation, Reconstruction and Animation in Forensic. Individual and Real 3D Data Based Geo-Metric Approach Including Optical Body/Object Surface and Radiological CT/MRI Scanning, J. Forensic Sci., 50, 2 (2005)

- [URL-1] Autodesk (Hrsg): 3DsMax.
<http://www.autodesk.de/store/products/3ds-max> (Zugriff am 28.08.2016)
- [URL-2] Gom (Hrsg): Industrielle 3D-Scanner.
<http://www.gom.com/de/messsysteme/atos.html> (Zugriff am 28.08.2016)
- [URL-3] Blender Foundation (Hrsg): About. <https://www.blender.org/> (Zugriff am 28.08.2016)
- [URL-4] Manuel Bastioni: Manuel Bastioni Laboratory.
<http://www.manuelbastioni.com/manuellab.php> (Zugriff am 28.08.2016)
- [URL-5] MakeHuman Team (Hrsg.): Make Human. <http://www.makehuman.org/> (Zugriff am 28.08.2016)
- [URL-6] Blender Wiki (Hrsg.): Archimesh.
https://wiki.blender.org/index.php/Extensions:2.6/Py/Scripts/Add_Mesh/Archimesh (Zugriff am 28.08.2016)
- [URL-7] Microvellum, Inc. (Hrsg.): Fluid Designer Download.
<http://www.microvellum.com/products/fluid-designer/fluid-designer-download/> (Zugriff am 28.08.2016)
- [URL-8] Github, Inc. (Hrsg.): antonioya/blender.
https://github.com/Antonioya/blender/blob/master/archimesh/archimesh_0_8_1.zip (Zugriff am 28.08.2016)
- [URL-9] Microvellum, Inc. (Hrsg.): Fluid Designer.
<http://www.microvellum.com/products/fluid-designer/> (Zugriff am 28.08.2016)
- [URL-10] Jean-Francois Gallant: CubeSurfer addon download.
<http://pyroevil.com/cubesurfer-addon-download/> (Zugriff am 28.08.2016)
- [URL-11] CgSpeed (Hrsg.): Motion Capture.
<https://sites.google.com/a/cgspeed.com/cgspeed/motion-capture> (Zugriff am 28.08.2016)
- [URL-12] Dennis Fassbänder: Fracture Modifier. <http://df-vfx.de/fracturemodifier/> (Zugriff am 28.08.2016)

-
- [URL-13] Acosta, Johnathan: Blend Swap. <http://www.blendswap.com/> (Zugriff am 28.08.2016)
- [URL-14] Roosendal, Ton: (Re-)defining Blender. 23. 10.2013 in: Roosendal, Ton: Blender developer blog. <https://code.blender.org/2013/10/redefining-blender/> (Zugriff am 28.08.2016)
- [URL-15] Roosendal, Ton: SIGGRAPH 2016 report. 7.08.2016, in: Blender Foundation (Hrsg.): Media Exposure. <https://www.blender.org/media-exposure/siggraph-2016-report/> (Zugriff am 28.08.2016)

Anlagen

Skript zum Import von Objekten:

Hierzu wird eine Objektbibliothek benötigt, die folgende Ordnerstruktur enthält:

Kueche – Buero – Andere – KuecheEssen - SchlafenWohnen

```
bl_info = {  
    "name": "Import Add-on",  
    "author": "Jasmin",  
    "version": (1,0),  
    "blender": (2,76,0),  
    "location": "View3D > Tools",  
    "description" : "Import Add-on für Möbel und diverse Objekte",  
    "category": "Object"  
}
```

```
import bpy  
import os
```

```
###-----UI Gestaltung-----###
```

```
-
```

```
###----- UI Part fuer Pfad Auswahl -----###
```

```
class directoryPan(bpy.types.Panel):
```

```
    bl_space_type = "VIEW_3D"
```

```
    bl_region_type = "TOOLS"
```

```
    bl_label = "Verzeichnis"
```

```
    bl_category = "Skript"
```

```
def draw(self, context):
```

```
    col = self.layout.column()
```

```
        #Zeichnen des Operators
```

```
        #Abkuerzungen festlegen
```

```
col.prop(context.scene, 'conf_path')          #Pfad-Auswahl
zeichnen ---- > verweist auf regis-ter
```

```
###----- UI Part fuer Kueche und Esszimmer -----###
```

```
class kuechePan(bpy.types.Panel):
    bl_space_type = "VIEW_3D"
    bl_region_type = "TOOLS"
    bl_label = "Kueche und Esszimmer"
    bl_category = "Skript"
```

```
def draw(self, context):
```

```
    col=self.layout.column()          #Abkuerzung
```

```
    obj_list = []      #erstellen einer Liste fuer die Objekte des Ordners
```

```
#----Schleife zum Finden der .obj-Dateien in den Ordner KuecheEssen ----#
```

```
    for obj in os.listdir(context.scene.conf_path + "KuecheEssen"):
```

```
#anfuegen des Ordners KuecheEssen and den Pfad
```

```
    if obj.split(".")[1] == "obj":
        #finden der .obj Dateien
```

```
        obj_list.append(obj.split(".")[0])
```

```
#Objektnamen an obj_list anfuegen und '.obj' abspalten
```

```
#--Schleife zum Erstellen der einzelnen Buttons fuer die Objekte-----#
```

```
    for i in obj_list:
        import_obj = "import." + i
```

```
col.operator('import.scanfile', text=i).objfile = i

#erstellen des operators
#import.scanfile verweist auf class ScanfileOperator mit dem bl_idname = im-
port.scanfile

###----- UI Part fuer label Buero -----###
class bueroPan(bpy.types.Panel):
    bl_space_type = "VIEW_3D"
    bl_region_type = "TOOLS"
    bl_label = "Buero"
    bl_category = "Skript"

    def draw(self, context):

        col = self.layout.column()

        obj_list = []

        for obj in os.listdir(context.scene.conf_path + "Buero"):
            if obj.split(".")[1] == "obj":
                obj_list.append(obj.split(".")[0])

        for i in obj_list:
            import_obj = "import." + i

            col.operator('import.scanfile3', text=i,
                        ).objfile = i

###----- UI Part fuer Schlafen und Wohnen -----###
class wohnenPan(bpy.types.Panel):
```



```
bl_space_type = "VIEW_3D"
bl_region_type = "TOOLS"
bl_label = "Schlafen und Wohnen"
bl_category = "Skript"

def draw(self, context):

    col = self.layout.column()

    obj_list = []

    for obj in os.listdir(context.scene.conf_path + "SchlafenWohnen"):
        if obj.split(".")[1] == "obj":
            obj_list.append(obj.split(".")[0])

    for i in obj_list:
        import_obj = "import." + i

        col.operator('import.scanfile2', text=i,
                    ).objfile = i
```

###----- UI Part fuer label Bad -----###

```
class badPan(bpy.types.Panel):
    bl_space_type = "VIEW_3D"
    bl_region_type = "TOOLS"
    bl_label = "Sanitär"
    bl_category = "Skript"

    def draw(self, context):

        col = self.layout.column()

        obj_list = []

        for obj in os.listdir(context.scene.conf_path + "Bad"):
```

```
if obj.split(".")[1] == "obj":
    obj_list.append(obj.split(".")[0])

for i in obj_list:
    import_obj = "import." + i

    col.operator('import.scanfile4', text=i,
                ).objfile = i

###----- UI Part fuer label Andere -----###
class anderePan(bpy.types.Panel):
    bl_space_type = "VIEW_3D"
    bl_region_type = "TOOLS"
    bl_label = "Andere"
    bl_category = "Skript"

    def draw(self, context):

        col = self.layout.column()

        obj_list = []

        for obj in os.listdir(context.scene.conf_path + "Andere"):
            if obj.split(".")[1] == "obj":
                obj_list.append(obj.split(".")[0])

        for i in obj_list:
            import_obj = "import." + i

            col.operator('import.scanfile5', text=i,
                        ).objfile = i
```

```
### ----- Definieren der Operator -----####

###-----Operator fuer Kueche und Esszimmer -----###
class ScanFileOperator(bpy.types.Operator):
    """Import an obj into the current scene"""

    #Beschreibung wenn man ueber den Operator hovert

    bl_idname = 'import.scanfile'

    #Verweist auf class kuechePan

    bl_label = 'Importiertes Objekt'

    #label im unteren Bereich der Toolbox

    bl_options = {'REGISTER', 'UNDO'} #Register-Optionen

    objfile = bpy.props.StringProperty(name="Objektname:")

    #Zugreifen auf StringProp, Zeigt den Namen im unteren Bereich
    #der Toolbox an, wenn man ein Objekt importiert

    #--Definieren was passiert wenn man auf einen Button klickt-----#
    def execute(self, context):
        print('importing', self.objfile)

    #Ausgabe in der Console, wenn darauf geklickt wird

    pfad = context.scene.conf_path + "KuecheEssen\\" + self.objfile+".obj"

    #zusammensetzen des Pfades

    bpy.ops.import_scene.obj(filepath= pfad) #Import des obj
    bpy.ops.object.origin_set(type = 'ORIGIN_CURSOR')
```

```
#Setzen der Origin zum 3DCursor
```

```
bpy.ops.object.origin_set(type = 'GEOME-TRY_ORIGIN')
```

```
#Setzen des Meshes zur Origin
```

```
return {'FINISHED'}
```

```
###-----Operator fuer Schlaf und Wohnzimmer-----###
```

```
class ScanFileOperator(bpy.types.Operator):
```

```
    """Import an obj into the current scene"""
```

```
    bl_idname = 'import.scanfile2'
```

```
    bl_label = 'Importiertes Objekt'
```

```
    bl_options = {'REGISTER', 'UNDO'}
```

```
    objfile = bpy.props.StringProperty(name="Objektname:")
```

```
    def execute(self, context):
```

```
        print('importing', self.objfile)
```

```
        pfad = context.scene.conf_path + "Schlafen-Wohnen\\"+ self.objfile+".obj" ###
```

```
        bpy.ops.import_scene.obj(filepath= pfad)
```

```
        bpy.ops.object.origin_set(type = 'ORIGIN_CURSOR')
```

```
        bpy.ops.object.origin_set(type = 'GEOME-TRY_ORIGIN')
```

```
    return {'FINISHED'}
```

```
###-----Operator fuer Buero -----###
```

```
class ScanFileOperator(bpy.types.Operator):
```

```
    """Import an obj into the current scene"""
```

```
    bl_idname = 'import.scanfile3'
```

```
    bl_label = 'Importiertes Objekt'
```

```
    bl_options = {'REGISTER', 'UNDO'}
```

```
objfile = bpy.props.StringProperty(name="Objektname:")
```

```
def execute(self, context):
    print('importing', self.objfile)
    pfad = context.scene.conf_path + "Buero\\"+ self.objfile+".obj"

    bpy.ops.import_scene.obj(filepath= pfad)
    bpy.ops.object.origin_set(type = 'ORIGIN_CURSOR')
    bpy.ops.object.origin_set(type = 'GEOME-TRY_ORIGIN')

    return {'FINISHED'}
```

```
###-----Operator fuer Sanitaer-----###
```

```
class ScanFileOperator(bpy.types.Operator):
```

```
    """Import an obj into the current scene"""
```

```
    bl_idname = 'import.scanfile4'
```

```
    bl_label = 'Importiertes Objekt'
```

```
    bl_options = {'REGISTER', 'UNDO'}
```

```
objfile = bpy.props.StringProperty(name="Objektname:")
```

```
def execute(self, context):
    print('importing', self.objfile)
    pfad = context.scene.conf_path + "Bad\\"+ self.objfile+".obj"

    bpy.ops.import_scene.obj(filepath= pfad)
    bpy.ops.object.origin_set(type = 'ORIGIN_CURSOR')
    bpy.ops.object.origin_set(type = 'GEOME-TRY_ORIGIN')

    return {'FINISHED'}
```

```
###-----Operator fuer Andere-----###
```

```
class ScanFileOperator(bpy.types.Operator):
```

```
'''Import an obj into the current scene'''
bl_idname = 'import.scanfile5'
bl_label = 'Importiertes Objekt'
bl_options = {'REGISTER', 'UNDO'}

objfile = bpy.props.StringProperty(name="Objektname:")

def execute(self, context):
    print('importing', self.objfile)
    pfad = context.scene.conf_path + "Andere\\" + self.objfile + ".obj"

    bpy.ops.import_scene.obj(filepath= pfad)
    bpy.ops.object.origin_set(type = 'ORIGIN_CURSOR')
    bpy.ops.object.origin_set(type = 'GEOME-TRY_ORIGIN')

    return {'FINISHED'}

def register():
    bpy.utils.register_module(__name__)

    #-----register fuer den Pfadoperator-----#

    bpy.types.Scene.conf_path = bpy.props.StringProperty \
    (
        name = "Pfad",
        default = "",
        description = "Verzeichnis festlegen",
        subtype = 'DIR_PATH'
    )

def unregister():
    bpy.utils.unregister_module(__name__)
    del bpy.types.Scene.conf_path
```

```
if __name__ == "__main__":  
    register()
```

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Ort, Datum

Vorname Nachname